

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE DE LA MAÎTRISE EN  
GÉNIE INDUSTRIEL

PAR

Walid Ghie

**Algorithme d'identification des chaînes de cotes en tolérancement  
assisté par ordinateur.**

**Juillet 99**

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

À ma future femme Mirvate

À ma famille

À ma ville natale Takrit (Liban)

*« Ô les gens ! Nous vous avons créés d'un mâle et d'une femelle, et nous vous avons désignés en nations et des tribus, pour que vous vous entre-connaissiez. Oui, le plus noble des vôtres, auprès de dieu, C'est le plus pieux des vôtres. Dieu est savant, informé, vraiment. » Coran (49-13)*

L'identification de chaînes de cotes fonctionnelles est une étape très importante pour le passage de la conception à la réalisation d'un assemblage. Chaque chaîne doit définir les exigences de qualité dimensionnelles et géométriques de fabrication des pièces juste nécessaire pour assurer leur bon fonctionnement tout en assurant l'assemblage facile du produit. Or, il n'existe actuellement aucun outil d'identification systématique des paires d'éléments fonctionnels affectant une condition jugée importante par le concepteur.

Le but de cette recherche est donc de développer un tel outil sous forme informatique. En particulier, à partir d'un dessin d'assemblage du produit en DAO, une structure de données menant à un modèle de type graphe (appelé graphe des éléments fonctionnels) est premièrement générée. Le concepteur identifie ensuite une paire d'éléments de ce graphe entre lesquels une condition jugée importante est à respecter. Un algorithme d'identification de toutes les combinaisons possibles de paires d'éléments formant un cycle autour de la condition à respecter est ensuite exécutée. Les éléments fonctionnels discutés ici sont en fait des points, des lignes et des surfaces, qu'ils soient fictifs ou réels.

L'algorithme proposé contient trois niveaux. Les premier et troisième niveaux sont plus étroitement liés aux moyens de définition géométrique (dessin DAO) et technologique des ensembles et des pièces mécaniques (affichage du résultat). Le deuxième niveau constitue pour sa part le noyau du traitement de la cotation fonctionnelle. Il contient trois modules. Son rôle est de générer de façon systématique le graphe de tous les éléments fonctionnels présents dans la représentation solide de l'assemblage tel que tiré d'une base de données d'un logiciel DAO. Son deuxième rôle est l'identification des surfaces de contacts géométriques entre les différentes pièces. À cette fin, l'utilisateur sélectionne la condition fonctionnelle et les différentes paires de surfaces en contact chacune appartenant à des pièces différentes. Après avoir élaboré le graphe des éléments fonctionnels il s'agit enfin d'identifier toutes les combinaisons possibles de paires d'éléments formant un cycle autour de la condition à respecter telle que sélectionnée par l'utilisateur, et de remonter de paire en paire d'éléments fonctionnels jusqu'à un retour au point de départ.

Cet algorithme permet ainsi de générer plusieurs chaînes tridimensionnelles autour d'une même condition. De plus, chaque paire d'éléments fonctionnels d'une chaîne peut impliquer des dispersions géométriques, i.e. l'algorithme identifie aussi les tolérances géométriques ayant un impact sur une condition fonctionnelle

# TABLE DES MATIÈRES

Résumé	I
Remerciements	II
Liste des tableaux	VI
Liste des figures	VII

## CHAPITRES

<b>I.</b>	<b>PROBLÉMATIQUE DE LA COTATION FONCTIONNELLE</b>	<b>1</b>
1.1.	ÉTUDE ANALYTIQUE DE LA COTATION FONCTIONNELLE	2
1.1.1.	Chaînes de cotes	2
1.1.2.	Intervalle de tolérance	5
1.2.	MOTIVATION	6
1.1.1.	Problématique de la cotation fonctionnelle	6
1.2.2.	Problématique d'intégration de la cotation fonctionnelle assistée par ordinateur	7
<b>II.</b>	<b>REVUE DE LA LITTÉRATURE</b>	<b>8</b>
2.1.	LES TOLÉRANCES	9
2.1.1.	Les tolérances dimensionnelles	9
2.1.2.	Les tolérances géométriques	10
2.2.	LES MODÉLISATIONS GÉOMÉTRIQUES	11
2.3.	LES DIFFÉRENTES APPROCHES PROPOSÉES POUR LE TOLÉRANCEMENT ASSISTÉ PAR ORDINATEUR	13
2.3.1.	Modèles de tolérancement par zones de tolérance	14
2.3.2.	Modèles de tolérancement par dimensions	16
2.3.3.	Modèle SATT	17
2.3.4.	Modèles statistiques	17
2.3.5.	Modèles généraux de tolérancement	18
2.4.	POSITIONNEMENT DE LA RECHERCHE	19
2.4.1.	Contexte	19
2.4.2.	L'approche préconisée	19
<b>III.</b>	<b>ALGORITHME D'IDENTIFICATION DES CHAÎNES DE COTES</b>	<b>21</b>
3.1.	ÉLÉMENTS DE BASE DE L'IDENTIFICATION DES CHAÎNES DE COTES	22
3.1.1.	Éléments fonctionnels (EFs)	23
3.1.1.1.	Définition	23
3.1.1.2.	Implantation	24
3.1.2.	Surfaces en contacts	28
3.1.2.1.	Définition	28
3.1.2.2.	Identification des surfaces en contact	28

3.1.3 Condition fonctionnelle (CF)	29
3.1.3.1 Définition	29
3.1.3.2 Identification d'une Condition fonctionnelle	30
3.1.4 Paires des éléments fonctionnels (PEFs)	31
3.1.4.1. Définition	31
3.1.4.2. Combinaisons des tolérances possibles	32
3.2 GRAPHE D'ÉLÉMENTS FONCTIONNELS (GEF)	32
3.2.1. Définition d'un graphe des éléments fonctionnels	32
3.2.2. Implantation du GEF	38
3.2.2.1. Représentation d'un GEF	38
3.2.2.2. Exploitation de la théorie de graphe	41
3.2.2.3. Algorithme d'implantation du GEF	42
3.2.3. Identification des paires des éléments fonctionnels	44
3.2.4. Affectation de tolérances sur une paire d'éléments fonctionnels	45
3.2.5. Existence et dénombrement des chemins dans un graphe	46
3.3 GÉNÉRATION DE L'ARBRE DES ÉLÉMENTS FONCTIONNELS (AEFS)	47
3.3.1. Définition	47
3.3.2. Représentation graphique	48
3.3.3. Implantation	50
<b>IV. ENVIRONNEMENT DU LOGICIEL</b>	<b>53</b>
4.1. LA STRUCTURE DU LOGICIEL	55
4.2. PRÉSENTATION GÉNÉRALE DU LOGICIEL.	55
4.2.1. Menu Fichier	56
4.2.1.1. L'option : Nouveau	56
4.2.1.2. L'option : Visualiser	58
4.2.2. Menu : Graphe d'éléments fonctionnels	59
4.2.2.1. L'option : Sélectionner des contacts	59
4.2.2.2. L'option : Sélectionner d'une condition fonctionnelle	61
4.2.2.3. L'option : Générer le graphe d'éléments fonctionnels	62
4.2.3. Menu : Arbre des paires d'éléments	63
4.2.3.1. L'option : Construction Arbre	63
4.2.3.2. L'option : Résultats des chaînes de cotes	64
4.2.3.3. L'option : Détail des chaînes de cotes	65
<b>V. EXEMPLE D'APPLICATION</b>	<b>66</b>
5.1. APPLICATION MANUELLE	68
5.1.1. Représentation du mécanisme à analyser	68
5.1.2. Éléments fonctionnels	69
5.1.3. Graphe d'éléments fonctionnels	70
5.1.4. Chaînes de cotes	75

5.2. CHEMINS POSSIBLES	76
5.3. RÉSULTATS DU PROGRAMME	77
5.3.1. Graphe d'éléments fonctionnels	77
5.3.2. Chaînes de cotes automatiques	78
5.3.3 Comparaison des différents résultats	78
<b>VI. CONCLUSION</b>	<b>80</b>
6.1 ACCOMPLISSEMENTS	81
6.1.1 Identification des éléments fonctionnels	81
6.1.2 Génération du graphe des éléments fonctionnels	81
6.1.3 Identification des cycles autour d'une condition fonctionnelle	82
6.1.4 Interface graphique d'utilisateur	82
6.2 UTILISATIONS POSSIBLES	83
6.2.1 Détermination des équations de l'intervalle de tolérance	83
6.2.2 Analyse de l'influence de chaque chaîne	84
6.2.3 Modélisation mathématique	85
6.2.4 Analyse de robustesse du design	86
6.2.4.1 Axiomes de robustesse	86
6.2.4.2 Application au tolérancement	87
5.3 LIMITES ET RECOMMANDATIONS	87
<b>RÉFÉRENCES</b>	<b>89</b>
<b>BIBLIOGRAPHIE</b>	<b>93</b>



## **Liste des tableaux :**

<b><u>Tableau</u></b>	<b><u>Titre</u></b>	<b><u>Page</u></b>
2.1	<i>Symboles des tolérances dimensionnelles</i>	9
2.2	<i>Symboles des tolérances géométriques</i>	10
2.3	<i>Les étapes de l'algorithme.</i>	20
3.1	<i>Algorithme de lecture du fichier BREP</i>	25
3.2	<i>Liste des éléments fonctionnels</i>	27
3.3	<i>Combinaisons des tolérances pour des paires d'éléments fonctionnels</i>	32
3.4	<i>Structure des données de sélections des contacts et C.F.</i>	43
3.5	<i>Algorithme de l'implantation de graphe d'éléments fonctionnels</i>	43
3.6	<i>Algorithme d'identification des paires d'éléments fonctionnels</i>	44
3.7	<i>Algorithme d'arbre d'éléments fonctionnels.</i>	51
5.1	<i>Matrice d'incidence initiale du GEF</i>	72
5.2	<i>Matrice d'incidence intermédiaire du GEF</i>	73
5.3	<i>Matrice d'incidence complète du GEF</i>	74
5.4	<i>Fichier résultats générer par le programme Analyse TAO</i>	78
6.1	<i>Équations d'intervalle de tolérances de l'exemple pompe à engrenages</i>	83

## Liste des figures :

<u>Figure</u>	<u>Titre</u>	<u>Page</u>
1.1	Chaîne de cotes d'une pièce.	3
1.2	Dimensions minimum et maximum.	4
1.3	Jmax en fonction de la variation des cotes contenantes et contenues.	4
1.4	Jmin en fonction de la variation des cotes contenantes et contenues	5
1.5	Mécanisme qui contient deux conditions fonctionnelles	6
2.1	Tolérance géométrique	11
2.2	Modélisation géométrique	12
2.3	Principe du décalage	15
2.4	Tolérancement vectoriel	16
3.1	Mécanisme tenant deux conditions fonctionnelles	22
3.2	Éléments fonctionnels d'un cylindre	23
3.3	Exemple d'un élément fonctionnel	23
3.4	Exemple d'un fichier BREP	24
3.5	Représentation de la liste des éléments fonctionnels en mémoire	26
3.6	Différentes pièces dans le montage de contre pointe	26
3.7	Exemple de composants en contact	28
3.8	Sélection des contacts des éléments fonctionnels	29
3.9	Condition fonctionnelle	30
3.10	Sélection d'une condition fonctionnelle	30
3.11	Exemple d'une notation d'une paire d'éléments fonctionnels	31
3.12	Exemple d'une paire d'éléments fonctionnels	31
3.13	Graphe d'éléments fonctionnels initial	34
3.14	Types de graphe A) Connexe B) Non connexe	34
3.15	Graphe des éléments fonctionnels intermédiaire	35
3.16	Graphe des éléments fonctionnels complet	36
3.17	Graphe des éléments fonctionnels «groupement par EFs »	37
3.18	Représentation par matrice d'incidence et d'adjacence	39
3.19	Représentation par structure d'adjacence	40
3.20	Représentation par matrice d'incidence	40
3.21	GEF par la matrice d'adjacence	41
3.22	Implantation d'un graphe d'éléments fonctionnels	42
3.23	Graphe d'éléments fonctionnels après l'affectation des tolérances	45
3.24	Le nombre de chemins possibles	47
3.25	Structure d'arbre binaire	48
3.26	Insertion dans une structure d'arbre binaire	49
3.27	Représentation graphique de l'arbre d'éléments fonctionnels	50
3.28	Exemple d'arbre binaire modifié	50
3.29	Arbre des paires d'éléments fonctionnelles de l'exemple	52
4.1	Structure interne du logiciel (Analyse TAO).	54
4.2	Écran d'accueil de (TAO)	55
4.3	Les différents sous menus dans (TAO)	55
4.4	Écran ouvrir un nouveau fichier.	56
4.5	Écran d'un nouveau fichier à analyser.	57
4.6	Écran visualisation d'un dessin.	58

4.7	<i>Écran Sélectionner des contacts.</i>	59
4.8	<i>Écran d'une suppression de contact.</i>	60
4.9	<i>Écran ajouter un contact.</i>	60
4.10	<i>Écran de sélection d'une condition fonctionnelle.</i>	61
4.11	<i>Écran de graphe d'éléments fonctionnels</i>	62
4.12	<i>Écran d'arbre d'éléments fonctionnels</i>	63
4.13	<i>Écran résultats des chaînes de cotes</i>	64
4.14	<i>Écran détail de la liste de chaîne des cotes</i>	65
5.1	<i>Mécanisme de la pompe à engrenage : arbre mené et arbre menant</i>	68
5.2	<i>Mécanisme de la pompe à engrenage : corps principal</i>	68
5.3	<i>Éléments fonctionnels : pompe à engrenage (Pièces #1 &amp; #2)</i>	69
5.4	<i>Éléments fonctionnels : pompe à engrenage (Pièces #3)</i>	70
5.5	<i>Graphe des éléments fonctionnels pour initial pour la pompe.</i>	71
5.6	<i>Graphe des éléments fonctionnels complet de la pompe</i>	74
5.7	<i>Résultats des tous les chemins possibles pour pompe à engrenage</i>	76
5.8	<i>Écran graphe d'éléments fonctionnels de l'exemple pompe à engrenage</i>	77
5.9	<i>Écran des chaînes de cotes de l'exemple pompe à engrenage</i>	78

# CHAPITRE 1

## PROBLÉMATIQUE DE LA COTATION FONCTIONNELLE

---

### 1.1 ÉTUDE ANALYTIQUE DE LA COTATION FONCTIONNELLE

1.1.1. Chaînes de cotes

1.1.2. Intervalle de tolérance

### 1.2 MOTIVATION

1.2.1. Problématique de la cotation fonctionnelle

1.2.2. Problématique d'intégration de la cotation fonctionnelle assistée par ordinateur

Le passage du dessin d'ensemble aux dessins de définition est souvent pour le concepteur une étape fastidieuse et pourtant déterminante pour la fabrication et le fonctionnement du produit. La cotation fonctionnelle est une fonction essentielle de ce passage. Elle doit définir les exigences de qualité dimensionnelles et géométriques de fabrication de pièces, juste nécessaire pour garantir les conditions du fonctionnement et de l'assemblage du produit.

Le but d'une cotation fonctionnelle est d'assurer le fonctionnement correct d'un mécanisme à partir de pièces fabriquées aux dimensions indiquées sur le dessin. Les cotes fonctionnelles expriment directement et sans ambiguïté les conditions requises pour l'aptitude à l'emploi de la pièce (notamment les conditions d'interchangeabilité). L'analyse technique d'une pièce permet de définir les fonctions à assurer et les conditions à remplir qu'exige le bon fonctionnement de cette pièce avec les autres pièces de l'ensemble dont elle fait partie.

## 1.1 ÉTUDE ANALYTIQUE DE LA COTATION FONCTIONNELLE

Dans la recherche des dimensions fonctionnelles d'une pièce, interviennent des considérations d'assemblage, de fonctionnement, de résistance, etc. Pour cette étude nous retiendrons celles qui paraissent les plus complexes en cotation, c'est-à-dire les considérations de l'assemblage et du fonctionnement.

Chaque fonction élémentaire d'un matériel est obtenue par un ensemble mécanique, formant ce qu'on peut appeler une chaîne cinématique, composée d'un certain nombre de pièces reliées entre elles par des joints fixes ou mobiles. Le calcul et l'étude d'un prototype montrent que la fonction n'est obtenue que dans la mesure où l'on assure, entre les surfaces terminales de cette chaîne cinématique, une certaine distance appelée jeu. [23][30]

### 1.1.1 Chaînes de cotes

Une surface fonctionnelle doit satisfaire une condition déterminée, qui peut avoir trait soit à la résistance de la pièce à laquelle elle appartient, soit à la liaison de cette pièce avec une pièce voisine.

Dans ce dernier cas, la condition se traduit par une dimension qu'on appelle **condition fonctionnelle**; on peut l'inscrire sur le dessin d'ensemble sous la forme d'une cote dont les lignes d'attache sont reliées à deux surfaces appartenant à deux pièces différentes, que l'on appellera **surfaces terminales**. La condition fonctionnelle est le premier élément d'une boucle fermée constituée par un système de cotes parallèles, partant des surfaces terminales, [23][24][30] et tel que :

- Chaque cote donne la distance entre deux détails d'une pièce;
- Chaque pièce apparaisse par une seule cote.

La boucle obtenue est donc une chaîne de cotes ; pour l'obtenir on part des surfaces terminales, deux lignes sur lesquelles on porte les cotes des pièces rencontrées, jusqu'à ce que les deux lignes aboutissent à une ligne d'attache commune qui définit la surface d'appui [29][30][35][43]. Un exemple fera mieux comprendre ce concept :

Considérons, la figure suivante, une pièce P1 d'ouverture «A» destinée à recevoir la pièce P2 d'épaisseur «B». La différence des cotes  $[A - B]$  est appelée **condition fonctionnelle** (jeu), et les

surfaces communes aux pièces « A » et « B » sont des **surfaces en contact**. L'introduction de la pièce « B » dans A s'effectuera aisément si la cote « A » est supérieure à la cote « B ». Le jeu ou la différence des cotes [A-B] traduit cette condition de montage.

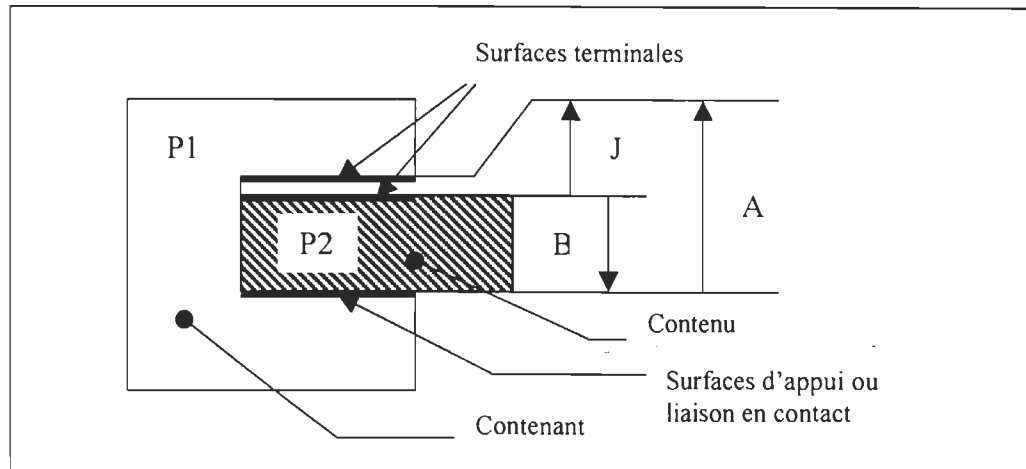


Figure 1.1 : Chaîne de cotes d'une pièce.

Les surfaces terminales appartiennent aux pièces "P1" et "P2". La chaîne de cotes devra donc commencer par une cote appartenant à la pièce "P1" et une cote appartenant à "P2". La chaîne de cotes de ce montage comprend les deux cotes, (A) la cote du contenant (P1) et (B) la cote du contenu (P2). Si nous considérons que les deux cotes sont fixes on trouve :

$$J = A - B \quad (1.1)$$

La relation précédente ne présente aucune difficulté. Mais une constatation expérimentale va faire disparaître cette simplicité, et donner naissance à de nombreuses complications :

**Une réalité :**

**Il est impossible de réaliser à l'atelier une dimension rigoureusement égale à une valeur donnée.**

On doit donc permettre à l'exécutant de s'éloigner de la cote nominale. Pour éviter tous les problèmes, on lui fixe deux limites entre lesquelles il doit tenir la dimension effective : on les appelle respectivement dimension maximale et dimension minimale [30][35][43] (Ces limites donnent à la pièce, dans la région considérée, le maximum de matière pour l'une, le minimum de matière pour l'autre). La pièce est acceptable si:

$$\text{Dimension minimale} \leq \text{Dimension effective} \leq \text{dimension maximale}$$

Alors, les deux cotes (A et B) sont variables entre deux valeurs minimum et maximum. Donc le jeu fonctionnel devrait être variable lui aussi. La figure 1.2 montre les schémas de ces deux pièces à leurs cotes maximum et minimum.

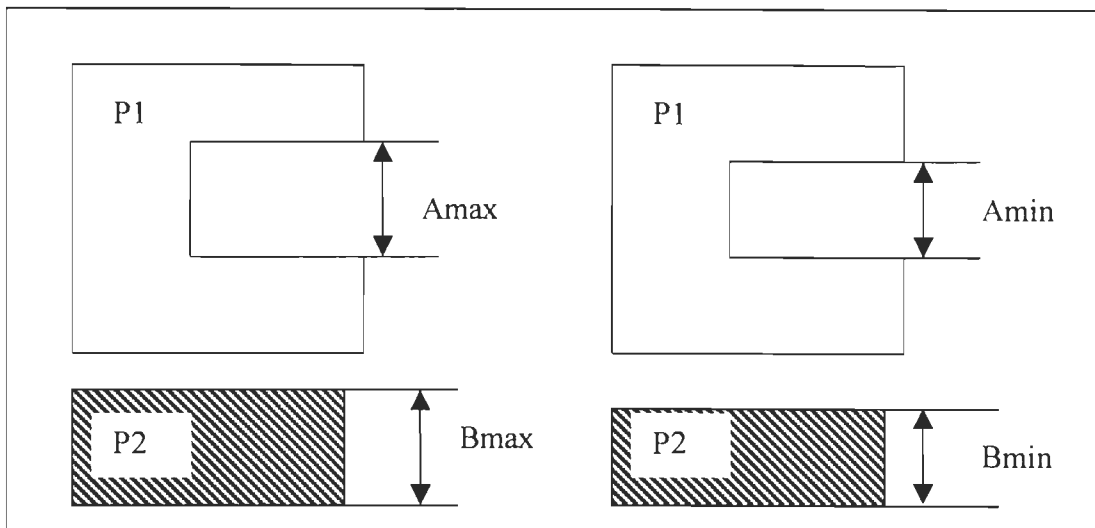


Figure 1.2 : Dimensions minimum et maximum.

La valeur maximum de J ( $J_{max}$ ) est obtenue avec la cote A maximum ( $A_{max}$ ) et la cote B minimum ( $B_{min}$ ). ( Figure 1.3)  $J_{max}$  devient :

$$J_{max} = A_{max} - B_{min} \quad (1.2)$$

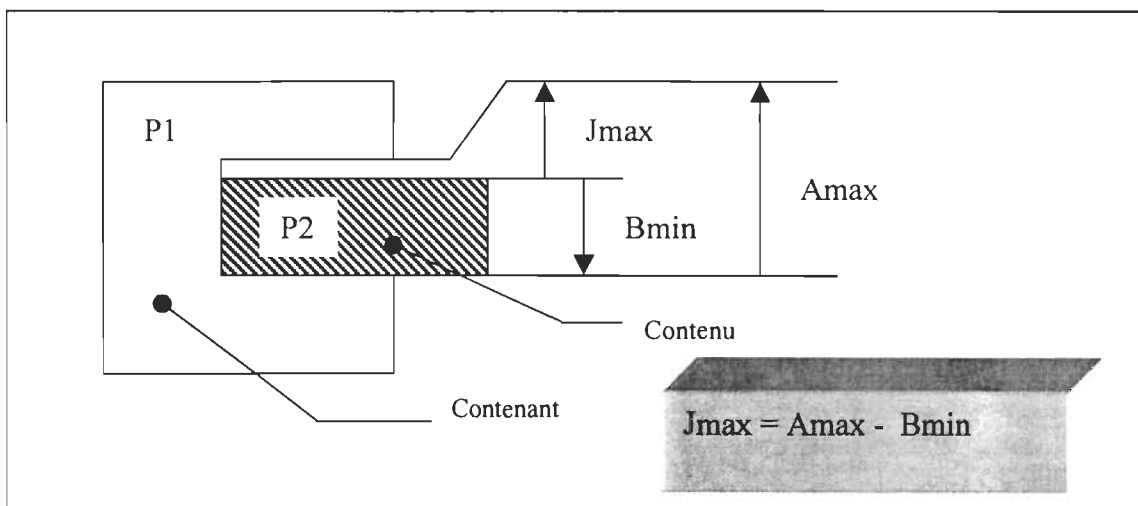


Figure 1.3 :  $J_{max}$  en fonction de la variation des cotes contenantes et contenues.

De façon identique, la valeur minimale de J ( $J_{min}$ ) obtenue avec la cote A minimum ( $A_{min}$ ) et la cote B maximum ( $B_{max}$ )  $J_{min}$  peut alors s'écrire :

$$J_{min} = A_{min} - B_{max} \quad (1.3)$$

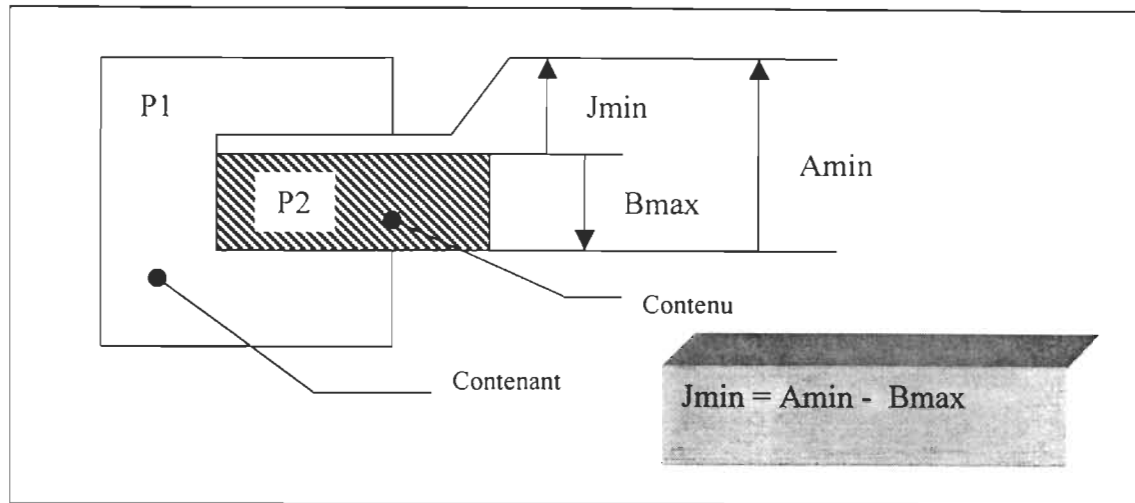


Figure 1.4 Jmin en fonction de la variation des cotes contenantes et contenues

Pour une cotation fonctionnelle avec jeu J, les expressions générales de Jmax et Jmin deviennent :

$$J_{\max} = \sum (\text{Contenantes maximum}) - \sum (\text{Contenues minimum}) \quad (1.4a)$$

$$J_{\min} = \sum (\text{Contenantes minimum}) - \sum (\text{Contenues maximum}) \quad (1.4b)$$

Ces deux équations (1.4a) et (1.4b) nous permettent de calculer l'intervalle de tolérance à imposer sur les différentes pièces de la chaîne des cotes obtenue. L'étude de cet intervalle de tolérance sera traitée en détail dans la section suivante.

### 1.1.2 Intervalle de tolérance

La condition fonctionnelle J doit être affectée d'un intervalle de tolérance (IT). Car il est impossible d'obtenir en fabrication des cotes constantes. Cette tolérance est choisie de manière à obtenir un jeu minimal et un jeu maximal compatible avec le fonctionnement correct. Cet intervalle de tolérance (IT) est ensuite réparti sur les cotes composant la chaîne des cotes, d'où découle le principe suivant. [24][30][33]:

L'intervalle de tolérance du jeu est égal à la somme de toutes les tolérances des cotes dans la chaîne de cotes.

$$IT(\text{jeu}) = \sum IT(\text{cotes}) \quad (1.5)$$

À première vue les relations précédentes sont faciles à appliquer, mais plus le nombre de conditions fonctionnelles augmente plus la complexité de ces équations augmente. Essayons d'appliquer ce principe sur un montage qui contient deux conditions fonctionnelles. Pour le montage de la figure 1.5, nous avons deux conditions fonctionnelles et pour chaque condition trois équations (1.4a), (1.4b) et (1.5); on voit facilement que la complexité augmente. Imaginez le cas où plus de cinq conditions fonctionnelles se manifestent. (Voir la figure 1.5 à la page suivante)



La figure 1.5 montre un mécanisme constitué de six pièces et de deux conditions fonctionnelles

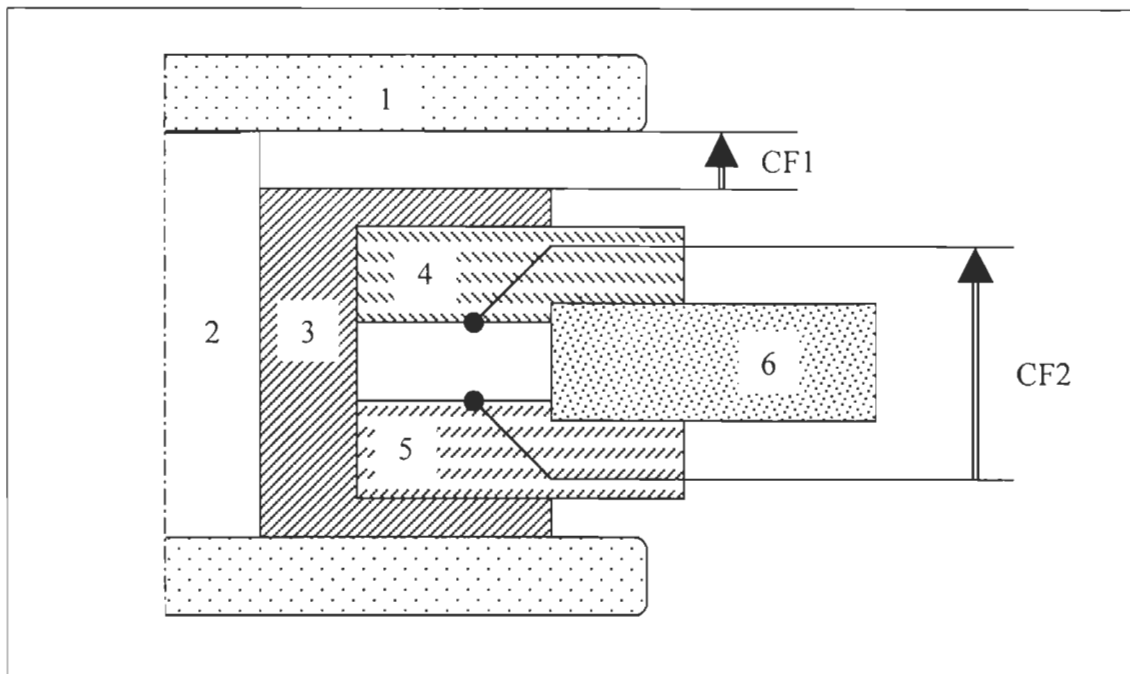


Figure 1.5 : Mécanisme qui contient deux conditions fonctionnelles

## 1.2 MOTIVATION

Les méthodes classiques de suivi de l'évolution des dimensions d'une pièce au cours de la gamme de fabrication apparaissent de moins en moins adaptées aux conditions actuelles de production industrielle en mécanique où l'intégration informatique est de plus en plus importante.

### 1.2.1 Problématique de la cotation fonctionnelle

Les conditions et les cotes fonctionnelles sont variables entre deux valeurs. Chaque condition introduit trois équations de résolution par chaîne fonctionnelle et trois inconnues pour chacune des  $N$  cotes composant la chaîne. La détermination des chaînes fonctionnelles doit par ailleurs prendre en compte les évolutions des contacts des états de fonctionnement imposés pour chaque condition fonctionnelle. Il apparaît de ce fait de nombreuses indéterminations dans le calcul des tolérances. De multiples contraintes sont d'autres part issues des conflits entre les différentes chaînes fonctionnelles, des conditions technologiques propres à chaque solution et des exigences de fabrication de chaque pièce, voire de chaque cote. L'ensemble de ces conflits peuvent être regroupés en trois classes : conflits dus aux ambiguïtés de conception, conflits issus des contraintes technologiques, et conflits issus des dépendances entre les chaînes fonctionnelles.[36]

Les origines des difficultés d'ordre méthodologique rencontrées dans la réalisation de la cotation fonctionnelle peuvent être résumées en quatre causes essentielles :

- Absence de méthode systématique de détection et de traitement des conflits de chaînes de cotes fonctionnelles; [2][9][10][36]
- Manque d'exhaustivité dû à la difficulté de prise en compte manuellement de toutes les conditions et contraintes de fonctionnement et montage dans des ensembles mécanique complexe; [29][31][35]
- Choix à priori de l'ordre de traitement de chaînes fonctionnelles masquant généralement la plupart des interactions entre les contraintes d'assemblage; [36]
- Choix des intervalles de tolérances pratiquement arbitraire. [30][31][41]

### 1.2.2 Problématique d'intégration de la cotation fonctionnelle assistée par ordinateur

De nombreux problèmes liés à la modélisation géométrique sont aujourd'hui résolus par la CAO, mais l'expérience montre que les systèmes actuels restent essentiellement dédiés à la «modélisation géométrique », autant en 2D qu'en 3D. Les principales difficultés de mise en œuvre de ces systèmes sont liées à la modélisation et au contenu des données existantes, pour différentes raisons :

- ☞ La modélisation géométrique ne porte que sur la définition géométrique de pièces isolées et parfaites; [3][4][7][23][36]
- ☞ La modélisation géométrique ne fournit pas les données fonctionnelles telles que les contacts entre pièces, l'association des conditions fonctionnelles et des cotes fonctionnelles à leurs surfaces et les évolutions des jeux en fonctionnement ; [5][6][11][36]
- ☞ L'extraction des données géométriques reste difficile [10][22][36];
- ☞ La réinjection directe des résultats du traitement de la cotation fonctionnelle, dans la représentation informatique et l'affichage des données des ensembles mécaniques, est difficile voire impossible [31][36].

En plus, les cotes et les tolérances dans les systèmes actuels de DAO sont sous la forme d'un texte, donc il n'y a aucun lien entre eux. L'une des contraintes importantes de la cotation est qu'elle doit respecter des normes. Les méthodes de cotation sont rarement automatiques. En général, l'utilisateur choisit au menu le type de cotation, montre l'objet ou les objets à coter, l'endroit où placer la flèche et divers renseignements (pré-texte, post-texte, tolérances...). Le système est alors capable de calculer la valeur de la cote en se référant au dessin et de dessiner l'ensemble des traits de rappel, flèche, textes. L'utilisateur peut ensuite modifier la position, la taille du texte ou la cote. À noter que les valeurs de la tolérance n'ont souvent aucune signification au niveau du dessin et que la modification de la valeur de la cote n'entraîne pas la modification du plan correspondant [31][35][36].

En conclusion, malgré la disponibilité sur le marché de multiples systèmes, leur lente évolution pour l'intégration des données fonctionnelles dans la modélisation des ensembles mécaniques conduit à ce que la réalisation de la cotation fonctionnelle reste, essentiellement, manuelle. Le processus de dessin (DAO) et conception (CAO) assistés par ordinateur impose de ce fait une phase manuelle qui constitue une rupture de la chaîne de CAO et pénalise sensiblement l'efficacité des systèmes industriels en construction mécanique.

# **CHAPITRE 2**

## **REVUE DE LA LITTÉRATURE**

---

### **2.1 LES TOLÉRANCES**

- 2.1.1. Les tolérances dimensionnelles
- 2.1.2. Les tolérances géométriques

### **2.2 LES MODÉLISATIONS GÉOMÉTRIQUES**

### **2.3 LES DIFFÉRENTES APPROCHES PROPOSÉES POUR LE TOLÉRANCEMENT ASSISTÉ PAR ORDINATEUR**

- 2.3.1. Modèles de tolérancement par zones de tolérance
- 2.3.2. Modèles de tolérancement par dimensions
- 2.3.3. Modèle SATT (Surfaces Associées Technologiquement et Topologiquement)
- 2.3.4. Modèles statistiques
- 2.3.5. Modèles généraux de tolérancement

### **2.4 POSITIONNEMENT DE LA RECHERCHE**

- 2.4.1. Contexte
- 2.4.2. L'approche préconisée

Depuis quelques années, la plupart des chercheurs, dans le domaine de tolérancement, ont emprunté le même cheminement pour proposer des modèles de traitement de la cotation fonctionnelle assistée par ordinateur, ce cheminement se résume en deux étapes :

1. L'analyse exhaustive sur les différents types de modélisation géométrique sur DAO;
2. L'analyse détaillée de la cotation fonctionnelle.

Ce chapitre a pour but de présenter les différents types de tolérances normalisées et les diverses méthodes des représentations géométriques. De plus on va donner un aperçu assez détaillé sur les différents modèles proposés par les chercheurs précédents.

## 2.1 LES TOLÉRANCES

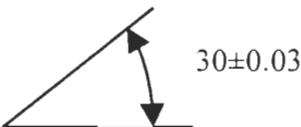
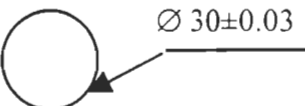
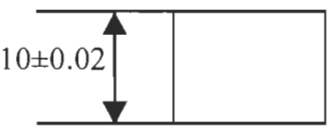
La forme définitive et les dimensions de l'objet fabriqué se représentent sur un dessin en décomposant ses éléments géométriques que l'on cote. Ces cotes sur le dessin donnent une description parfaite de la pièce en tous points. [25][30]

Or, l'imprécision inévitable des procédés d'usinage fait qu'une pièce ne peut pas être réalisée d'une façon rigoureuse et conforme aux dimensions fixées au préalable. Il a donc fallu tolérer que la dimension effectivement réalisée soit comprise entre deux dimensions limites, compatibles avec le fonctionnement correct de la pièce. La différence entre ces dimensions constitue la **tolérance**. Au sens général du terme, la tolérance est l'acceptation par le contrôle final d'une pièce considérée, dont tous les éléments de deux étalons fictifs parfaits (étalons de référence) sont établis dans le maximum et minimum de matière admissible. Cette «tolérance» générale se divise en plusieurs tolérances particulières. Elles sont caractérisées principalement par deux catégories importantes : tolérances dimensionnelles et tolérances géométriques. [30]

### 2.1.1 Les tolérances dimensionnelles

Théoriquement les tolérances dimensionnelles sont utilisées chaque fois que l'on porte une dimension (cote) sur le dessin d'une pièce. Pratiquement, seules quelques cotes dimensionnelles principales sont tolérancées, les autres pouvant obéir à une tolérance générale. Ces types de tolérances affectent directement et uniquement les dimensions, en occurrence les dimensions linéaires, radiales et angulaires. [34]

Tableau 2.1 : Symboles des tolérances dimensionnelles

Tolérance angulaire	Tolérance radiale	Tolérance linéaire
		







Les tolérances dimensionnelles s'expriment par  $G \pm t$  avec  $G$  dimension nominale et  $2t$  intervalle de tolérance. [30][34]

### 2.1.2 Les tolérances géométriques



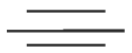



On a vu que les dimensions d'une pièce sont affectées de tolérances dimensionnelles. Ainsi pour le diamètre d'un cylindre on définit deux limites, respectivement au maximum et minimum de matière. Toute pièce réalisée entre ces deux limites sera acceptée par les appareils de contrôle. Ceci ne veut pas dire qu'elle soit pleinement satisfaisante. On remarque en effet dans l'aspect géométrique de l'objet des défauts concernant une seule propriété (cylindricité, rectitude, planéité,.. ), soit une relation entre deux éléments (parallélisme, perpendicularité, ..). On appelle les premiers, défauts de forme et les autres défauts d'orientation et de position. Tous ces défauts ont une influence sur le contact entre les pièces voisines.

Pour assurer des conditions correctes de fonctionnement et d'aptitude à l'emploi des produits finis, les normes précisent les principes de symbolisations et d'indication des tolérances géométriques sur les dessins techniques [10][23][24][30][31]. Les différents symboles des tolérancements géométriques sont présentés dans le tableau 2.2.



Tableau 2.2 : Symboles des tolérances géométriques

Tolérances de formes	Symboles	Tolérances de formes	Symboles
Cylindricité (EI)		Planéité (EI)	
Ligne quelconque (EI/EA)		Surface quelconque (EI/EA)	
Circularité(EI)		Rectitude (EI)	

Tolérances de position (EA)	Symboles	Tolérances d'orientation(EA)	Symboles
Coaxialité, concentricité		Parallélisme	
Symétrie		Perpendicularité	
Localisation		inclinaison	

Tolérances de battement(EA)	Symboles	Tolérances de battement(EA)	Symboles
Battement simple		Battement total	

On remarque ici qu'on a deux types de tolérances géométriques : l'une s'applique sur un élément isolé (EI), l'autre s'applique sur deux éléments associés (EA) et l'un se réfère à l'autre. Toutes ces tolérances doivent être situées à l'intérieur des zones de tolérances dimensionnelles [31]. Ces différentes zones de tolérancement sont illustrées à la page suivante.

Ces tolérances sont décrites dans les normes. Cependant elles n'ont pas fait l'objet d'une définition générique et restent par conséquent souvent perçues comme étant limitées aux cas pris en exemple dans les normes.

Une tolérance géométrique comporte :

- ♦ Des éléments tolérancés,
- ♦ Des zones de tolérances,
- ♦ Des références spécifiées.

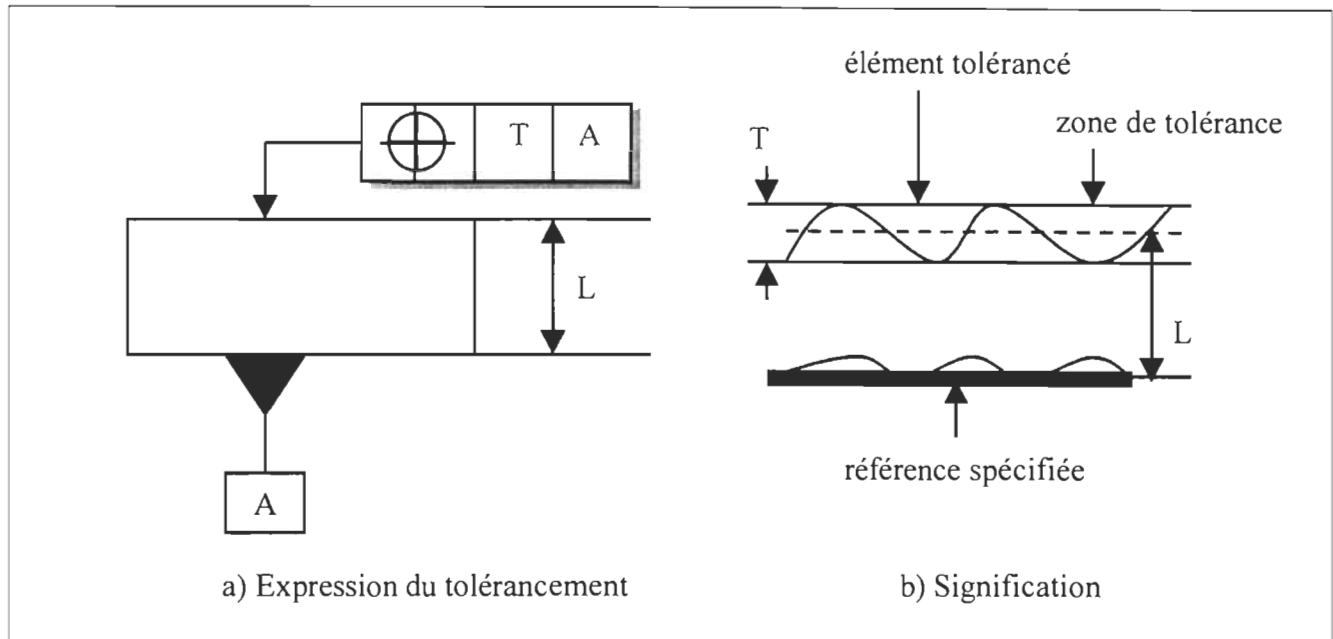


Figure 2.1 : Tolérance géométrique

Les éléments tolérancés sont soit des éléments géométriques réels soit des éléments géométriques fictifs. Les zones de tolérances sont des espaces surfaciques ou volumiques dans lesquels doivent se trouver les éléments tolérancés. Enfin, les références spécifiées sont des éléments géométriques supposés parfaits contraints en position et/ou en orientation et associés à des éléments réels. Elles orientent et positionnent les zones de tolérances.

## 2.2 Les modélisations géométriques

Le terme de : « modèle géométrique » est entendu au sens d'un ensemble d'outils mathématiques (en particulier géométrie analytique) permettant de définir (géométriquement et topologiquement), dans la machine, la forme de ce qui sera un objet ou un ensemble d'objets matériels (après fabrication). Dans cette section nous allons présenter en premier lieu les différentes modélisations géométriques existantes dans les logiciels de DAO, ensuite celles qui sont utilisées pour les traitements du tolérancement assisté par ordinateur. [22][38][39][40]

À l'origine, dans les années 70, les systèmes de DAO/CAO étaient des programmes d'application très spécialisés dans des secteurs d'activités précises (mécanique, bâtiment, électronique) et la nature des informations traitées étaient très fortement liées à l'application. Au fur et à mesure que les systèmes ont évolué pour devenir plus généraux les problèmes mathématiques et informatiques ont été résolus et la puissance des ordinateurs a augmenté, et ainsi de nouveaux modèles géométriques ont été développés. Ces modèles peuvent être classés en trois classes différentes (Figure 2.2) : Modélisation filaire, surfacique et volumique. [10][22][40][42]

La figure 2.2 montre les différents modèles de modélisation en DAO

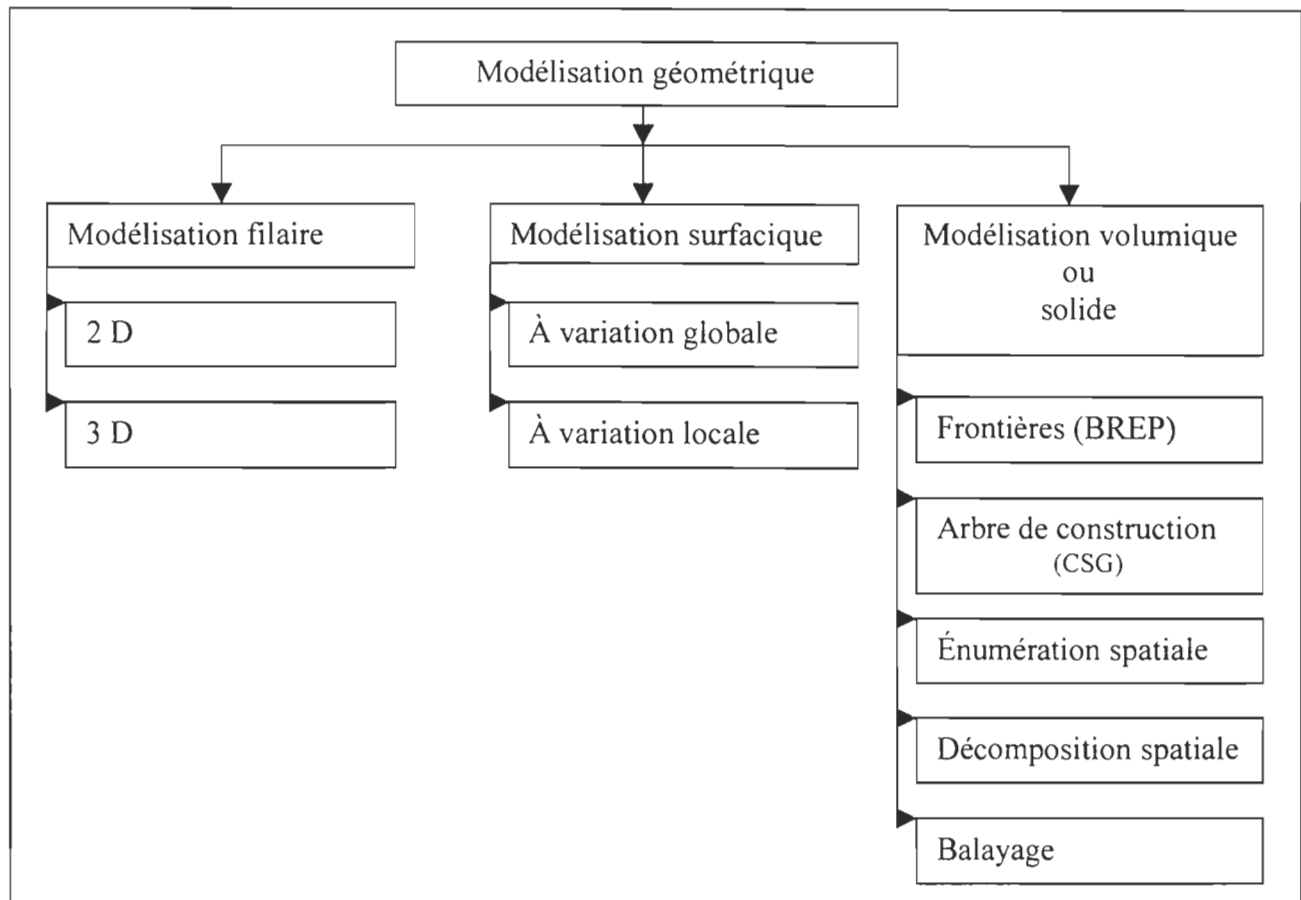


Figure 2.2 Modélisation géométrique

La définition de chaque type de modélisation sera présentée aux pages suivantes.

### Modélisation filaire [10][22][42]

Modélisation filaire (fil de fer, «wire frame») : Directement inspiré du dessin industriel, dans ces modèles l'objet n'est représenté que par ses contours. On peut distinguer les systèmes travaillant en deux dimensions et en trois dimensions. Ces systèmes manipulent des entités telles que le point, le segment de droite, l'arc de cercle et d'ellipse ainsi des courbes planes paramétriques comme les Beziens ou les Splines. **Les modèles filaires sont loin de posséder les propriétés souhaitables d'un bon modèle géométrique.**

### Modélisation surfacique [10][22][42]

Ces modèles ont été élaborés pour répondre aux besoins très spécifiques des industries de l'automobile et de l'aéronautique. Ces entreprises sont confrontées au traitement des surfaces gauches résultant de contraintes mécanique, d'aérodynamique. Plusieurs formalismes mathématiques ont été imaginés dont les plus couramment utilisés sont : les surfaces de Cons, Bézier, Splines, Bspline et NURBS. Certains formalismes sont dits à variation globale car la modification d'un

sommet quelconque du réseau a une influence sur toute la surface (Bézier). D'autres sont dits à variation locale car ils ne prennent en compte qu'un nombre limité de points (fonction du degré de la courbe) pour définir une proportion de la surface localement (B-Spline). **Ces modèles ne s'intéressent qu'à la peau des objets et ne possèdent pas toutes les propriétés souhaitables d'une bonne représentation géométrique.**

### Modélisation volumique ou solide [10][22][42]

La modélisation du solide (parfois appelée modélisation volumique) a été introduite en informatique graphique et en CAO pour surmonter les limitations de la modélisation surfacique

Ces modèles permettent de représenter de manière exacte ou approchée le volume d'un objet, c'est-à-dire que pour chaque point de l'espace, le système fait ou non partie de l'objet. Ces modèles possèdent de belles qualités mais exigent une capacité mémoire importante ce qui rend leur utilisation difficile.

Comme pour le précédent (modélisation surfacique), ce modèle définit l'objet plutôt que les projections de l'objet sur des plans. Mais à la différence du précédent, on définit une enveloppe fermée, c'est-à-dire un volume. On recense plusieurs principes de modélisations pour définir un volume (Figure 2.1), les plus utilisés sont la modélisation par arbre de construction (CSG) et la modélisation par les limites (BREP).

### Le modèle CSG [10][22][42]

L'arbre de construction (Constructive Solid Geometry) est obtenu à partir d'un ensemble de paramètres primitifs (boîtes, prismes, cylindre, cônes, sphères) sur lesquels sont appliqués des opérateurs ensemblistes (union, intersection, différence) et des transformations géométriques (translation, rotation, duplication, changement d'échelle). Le modèle est donc stocké sous forme d'une phrase qui décrit l'historique de la création. Cette phrase décrit l'arbre de construction.

### Le modèle par les limites ou BREP [10][22][42]

La représentation par frontières est l'héritière directe de la modélisation surfacique, mais elle garantit la cohérence topologique. L'objet est décrit par son contour : ses facettes, ses arcs ou arêtes, ses sommets, et les diverses relations topologiques entre ces éléments du contour. Les sommets, arêtes et faces sont usuellement décrits par des enregistrements et leurs relations par références (typiquement des pointeurs ou des indices dans des tables) entre ces enregistrements.

La représentation BREP se différencie par l'étendue de leurs topologies, et par le type et le degré de leurs surfaces.



## 2.3 LES DIFFÉRENTES APPROCHES PROPOSÉES POUR TOLÉRANCEMENT ASSISTÉ PAR ORDINATEUR

L'utilisation systématique de l'ordinateur à toutes les étapes du cycle de vie des produits et plus particulièrement au niveau de la conception, de la fabrication et du contrôle, nécessite le besoin de disposer entre autre d'un modèle cohérent de définition de la géométrie fonctionnelle. On constate que les systèmes informatiques de C.F.A.O. disponibles aujourd'hui ne répondent pas à ce besoin. Ils ne permettent de définir qu'une géométrie idéale tridimensionnelle certes exploitable à diverses étapes de l'élaboration des produits mais insuffisante puisqu'elle n'intègre pas les aspects fonctionnels. La solution actuelle pour transmettre les informations géométriques fonctionnelles liées aux produits reste alors très conventionnelle, elle passe par la réalisation des dessins techniques. Bien que très fidèle à la culture technologique du monde de la mécanique, cette solution présente des ambiguïtés liées au langage graphique normalisé employé. En effet, les tolérances qui traduisent l'aspect fonctionnel des produits ne respectent pas le principe d'univocité et ne permettent pas d'exprimer toutes les exigences fonctionnelles. Ce constat n'est pas récent, des équipes de recherche à travers le monde se mobilisent depuis environ une dizaine d'années et le nombre d'articles traitant de ce problème est important. Toutefois, ces derniers temps, on peut remarquer une recrudescence des activités.

Au niveau européen, sous l'impulsion du danois Per Bennich, le C.E.N. a cherché à harmoniser les normes traitant du tolérancement et de la métrologie des produits [34].

Au niveau international, les travaux de normalisation européens ont été proposés à l'ISO pour être approfondis et normalisés. En recherche, le Collège International pour l'Etude Scientifique des Techniques de Production Mécanique (C.I.R.P.) organise tous les deux ans depuis 1989 un séminaire international sur le tolérancement. Le dernier ayant pour titre «Computer Aided Tolerancing » s'est tenu à Enschede 1999.

Enfin au niveau américain, sous l'égide de l'American Society for Mechanical Engineering (A.S.M.E.) et du National Science Foundation (N.S.F.) un groupe de travail s'est réuni en 1988 pour traiter du sujet « Research needs and technological opportunities in mechanical tolerancing ». Les travaux se sont traduits par la tenue à Dearborn en 1993 d'un forum international ayant pour titre « Dimensional Tolerancing and Metrology » [34].

Tous ces travaux et activités ont produit des résultats qui contribuent à satisfaire le besoin défini ci-dessus. Cependant leur analyse fait apparaître des solutions qui varient en fonction du point de vue des auteurs. De manière générale, on distingue ceux qui cherchent d'autres formes de langage d'expression des exigences fonctionnelles ou qui cherchent l'amélioration du langage normalisé des tolérances, ceux qui s'appuient sur les modeleurs géométriques utilisés dans les systèmes de C.F.A.O. pour intégrer les tolérances et construire des outils d'analyse et de synthèse, enfin ceux qui élaborent un modèle basé sur les méthodes d'identification géométrique des produits en métrologie tridimensionnelle sans se soucier des acquis en modélisation et en normalisation mais assurant la cohérence pour tous les acteurs de l'entreprise.

### 2.3.1 Modèles de tolérancement par zones de tolérance

Le modèle de A. A. et G. Requicha [2] est construit sur une modélisation solide de type CSG. L'auteur considère que la sémantique des tolérances doit être définie mathématiquement pour éviter

toutes les ambiguïtés d'interprétation et pour assurer la cohérence avec les algorithmes de traitement en métrologie. Sa théorie repose sur la création d'une zone de tolérance par décalage de la géométrie nominale «offset zone» [8].

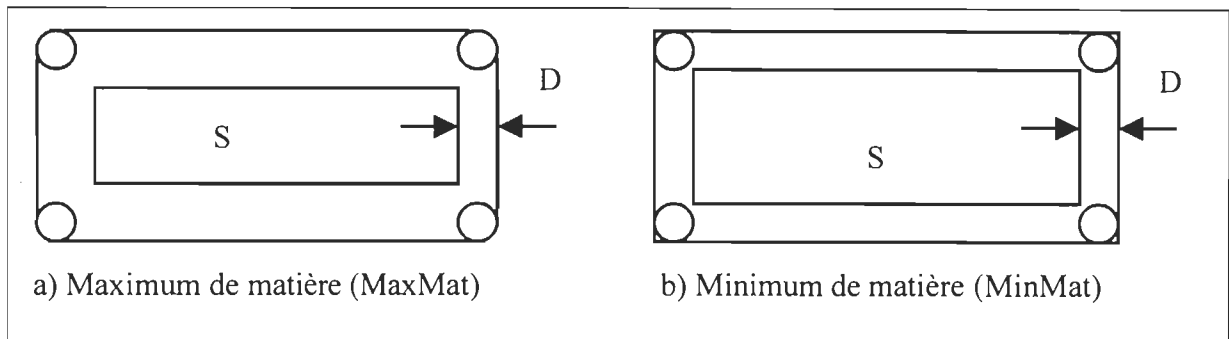


Figure 2.3 : Principe du décalage

Les zones décalées sont construites à l'aide des opérations de Minkowski illustrées par la figure 2.3 sur un problème plan pour simplifier la compréhension. La frontière MaxMat (a) est obtenue par déplacement d'un cercle de diamètre  $D$  à l'extérieur du contour. La frontière MinMat (b) est obtenue de la même manière avec le cercle à l'intérieur du contour. La soustraction des deux surfaces obtenues fournit la zone de tolérance.

La forme du contour décalé est donnée par des équations du type  $O(D,S) = \{p; \text{dis}(p,S) \leq D\}$  (chacune de ces équations est en fonction de deux paramètres  $D$  et  $S$  pour assurer une zone de tolérance acceptable,  $\text{dis}$  est une fonction de distance). A partir de cette théorie, l'auteur définit des classes d'objets appelées «variational class» qui sont utilisées pour représenter une famille d'objets physiques similaires à l'objet nominal et restant fonctionnellement équivalents. Ce modèle considère uniquement les tolérances par zones, rien n'est dit sur les éléments réels et sur l'association des éléments parfaits aux éléments réels. Cette modélisation ne convient pas aux tolérances dimensionnelles et aux tolérances de position normalisées.

Requicha a étudié un autre modèle basé sur la variation des paramètres de définition de la géométrie nominale appelé «parametric tolerancing». Pour lui, ce modèle n'est pas satisfaisant puisqu'il ne prend en compte que des éléments géométriques parfaits loin de la réalité.

Etesami a mis en évidence la non conformité de la théorie de Requicha avec la norme ANSI Y14.5 et propose un langage de spécification des tolérances géométriques (Tolerancing Specification Language, TSL). Ce langage est basé sur un ensemble de théories et en particulier sur le concept d'offset. Il n'est pas limité en types d'éléments et ne considère aucune classification des tolérances. La géométrie nominale est modélisée par des demi-espaces  $f(p) \leq 0$ . A un élément réel modélisé par un sous-ensemble de  $E^3$  correspond un élément nominal. La relation entre un élément nominal et un élément réel est définie par des fonctions de distances. [34]

Ce langage s'applique aux surfaces et considère des zones de tolérances uniformes. Srinivasan a étendu les possibilités proposées par Requicha en complétant le concept de décalage par celui de balayage pour pouvoir traiter les tolérances dimensionnelles et de position normalisée [47]. Le principe consiste à définir une zone par déplacement d'une sphère dont le centre est astreint à se trouver sur un point, sur une ligne ou sur une surface définie. [34]

### 2.3.2 Modèles de tolérancement par dimensions

Le modèle de A. Wirtz baptisé "tolérancement vectoriel" s'appuie sur la géométrie vectorielle pour définir les éléments géométriques constitutifs d'une pièce. Quatre vecteurs décrivent, dans un repère unique, la géométrie nominale de chaque élément. Ces vecteurs représentent la position, l'orientation, la dimension et la forme d'un élément [10][34]. ( Figure 2.4)

La variation de la géométrie est exprimée par la variation des composantes de ces quatre vecteurs. Ce modèle répond à l'objectif de cohérence recherché puisqu'il est unique pour tous les acteurs de l'entreprise. Cependant, il s'éloigne très franchement du langage de tolérancement normalisé et par conséquent ne permet pas d'avoir une expression fidèle des aspects fonctionnels. Il est plus proche des besoins de la fabrication et du contrôle que de ceux des études.

La figure suivante illustre le principe du modèle tolérancement par dimensions

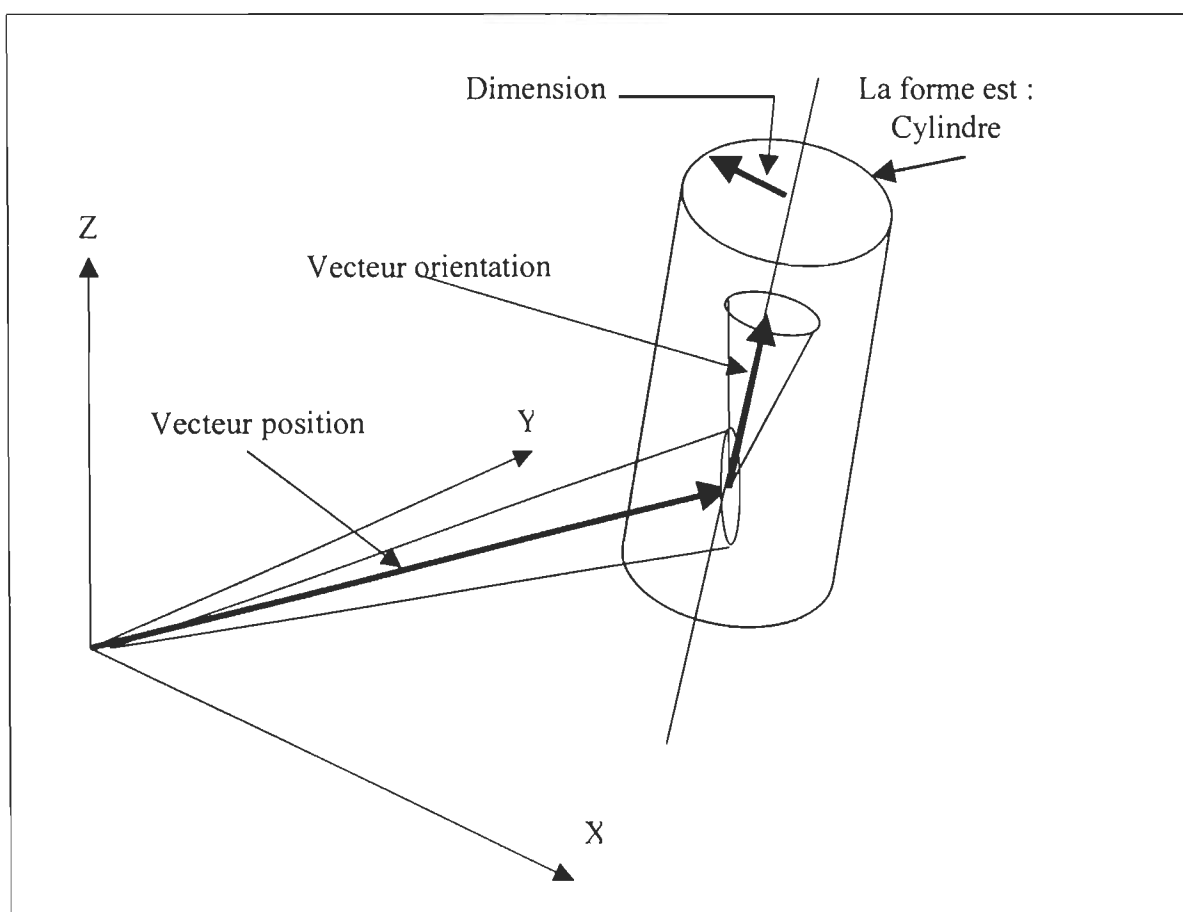


Figure 2.4 : Tolérancement vectoriel

### 2.3.3 Modèle SATT (Surfaces Associées Technologiquement et Topologiquement)

Gaunet propose également une approche du tolérancement vectoriel mais assez différente de celle de Wirtz . Le modèle développé repose sur le concept de Surfaces Associées Technologiquement et Topologiquement (SATT) défini par A. Clément. Ce modèle est construit sur le groupe et les sous-groupes des déplacements qui permettent de classer les surfaces fonctionnelles de la mécanique. Les auteurs considèrent 7 classes de surfaces. L'association de ces surfaces pour traduire les relations fonctionnelles dans le mécanisme permet de définir un SATT qui est lui-même reclassé dans une classe de surface. La situation relative des SATT est décrite par un paramétrage linéaire et angulaire choisi entre les Éléments Géométriques Minimums (EGRM) de chaque surface. La variation de la géométrie nominale ainsi décrite est représentée par des incertitudes sur les valeurs de ces paramètres et exprimée par des torseurs de petits déplacements. Cette modélisation offre des possibilités exploitées par les auteurs pour définir des outils de synthèse et d'analyse du tolérancement dans les mécanismes. L'expression du tolérancement proposée s'appuie sur le langage normalisé des tolérances géométriques par zones. Le passage du modèle proposé à l'expression du tolérancement par zones n'est pas explicite. Les critères d'association des éléments parfaits aux éléments réels ne sont pas pris en compte dans le modèle. [5][6][13][19]

### 2.3.4 Modèles statistiques

Plusieurs chercheurs se basent sur les statistiques pour résoudre le problème de tolérancement. Trois approches sont connues dans ce domaine : *programmation linéaire*, *linearisation directe DLM* (*Direct Linearization Method*) et celui de Monte Carlo.

#### *Programmation linéaire :*

Cette approche a été introduite par Turner et a été améliorée par Peter Martino. Dans cette approche, les éléments géométriques et les tolérances sont représentés par des contraintes et les degrés de liberté des corps en mouvement et les dimensions sont des variables dans la formulation de la programmation linéaire. Ji (1993) a proposé un modèle de programmation linéaire pour l'affectation de tolérances sur une chaîne des cotes. [8][46]

#### *Linearisation directe DLM (Direct Linearization Method) :*

Kenneth W. Chase, Spencer P., Magleby Jinsong Gao proposent une méthode statistique qui consiste à estimer quantitativement la variation de toutes les caractéristiques critiques de l'assemblage et prévoit le pourcentage des assemblages qui ne respecteront pas les spécifications de la conception. Dans cette approche les contraintes cinématiques sont utilisées comme des fonctions de l'assemblage. Pour les boucles de vecteur fermées, le début et la fin des bases de référence doivent être situés au même point et être orientés dans la même direction quand chaque vecteur de la boucle a été traversé. Cela nous permet d'égaliser le produit des transformations (translation et rotation) au vecteur unité. [26][45]

#### *L'algorithme de Monte Carlo :*

Les méthodes de Monte Carlo sont des méthodes probabilistes basées sur l'utilisation de suites aléatoires indépendantes uniformément distribuées. Elles sont généralement utilisées pour des problèmes que les méthodes classiques n'arrivent pas à résoudre.[45]

D'autres techniques peuvent être utilisées telles que les techniques dites de «quasi-Monte Carlo». Ces méthodes sont analogues à celles plus connues de Monte Carlo, mais au lieu d'utiliser une suite de variables aléatoires indépendantes uniformément distribuées, elles utilisent des suites déterministes, dites «à discrétance faible ». Ces suites ont l'avantage d'être très bien réparties sur la chaîne des cotes. [26][45]

### 2.3.5. *Modèles généraux de tolérancement*

Il existe d'autres modèles de tolérancement que nous allons énumérer par la suite :

- ◆ Stewart propose une théorie qui intègre le tolérancement par zones de tolérance et par dimension sur la base de la théorie de Requicha. [10][34];
- ◆ Parkison [41][46] propose une approche basée sur les statistiques pour l'analyse des tolérances ainsi que pour la distribution de ces tolérances en tenant compte des procédés de fabrication. Il a développé une formulation ou un modèle mathématique programmable en prenant comme variable objective le coût du produit et comme variable décisionnelle la déviation standard du design. De son analyse, il est ressorti que plus les cotes sont élevées plus les produits sont rejetés et plus les coûts sont élevés. Par contre cette méthode est applicable pour les lots de pièces[41];
- ◆ Xiaoqing et Davies (1988) [46] ont proposé la méthode «Matrix tree chain » pour identifier les chaînes dimensionnelles et assigner les tolérances sur la base du traitement itératif. La méthode est développée dans l'objectif d'automatiser le tolérancement dimensionnel
- ◆ La technique (Tolerance Charting ) de construction de diagramme de tolérance a été employée par les ingénieurs des méthodes dans l'industrie pour calculer les dimensions et les tolérances fonctionnelles dans la planification du processus de fabrication. [11];
- ◆ Ngoi et Teck (1993) ont proposé une méthode pour automatiser la détermination des dimensions fonctionnelles et leurs tolérances équilibrées. La méthode est basée sur une technique traçante de voie d'accès en utilisant une approche graphique ensuite la programmation linéaire;
- ◆ Serge Samper, Max Giordano [44] proposent d'intégrer les déformations élastiques mécaniques dans la démarche de tolérancement. Quatre modèles par ordre de complexité étaient proposés. Le premier est un modèle de «mécanisme parfait » qui permet une analyse cinématique et statique classique. Le second est un modèle «mécanisme rigide avec jeux et écarts » qui permet une première approche du tolérancement. Le dernier modèle est une synthèse des deux précédents qui rend possible la prise en compte des déformations dans la détermination des tolérances. L'hypothèse que les pièces sont rigides est rejetée selon les auteurs des ces modèles;
- ◆ P. Bourdet a développé un modèle ( $\Delta I$ ) géométrique pour le suivi de la cotation en cours de fabrication. Le but de cette méthode est de valider une gamme, mais l'application de cette méthode reste lourde. Elle est souvent traitée manuellement, et de façon incomplète. Par exemple, les variations de dimensions associées à des processus complexes comme les traitements spéciaux sont fréquemment négligés. J.F. Rigal, V. Wolff, B. Roussés, S. Raynaud proposent une extension de la méthode de ( $\Delta I$ ) de P. Bourdet aux spécifications géométriques en vue de son intégration dans l'ensemble des logiciels de CAO. [27]

Le sujet de l'analyse et de la synthèse des tolérances reste ouvert sur plusieurs points :

- ◆ Calcul des tolérances par minimisation du coût global du produit,
- ◆ Prise en compte des tolérances pour la conception robuste des systèmes mécaniques,
- ◆ Intégration de l'analyse et de la synthèse dans les systèmes de CFAO.

Actuellement, ces trois points sont traités de façon informelle (semi-automatique/semi-manuelle), grâce au savoir-faire et à l'expérience des concepteurs et des fabricants. Le passage à une approche optimale et automatique fait partie d'objectifs à long terme.

## **2.4 POSITIONNEMENT DE LA RECHERCHE**

### **2.4.1. Contexte**

La détermination des tolérances géométriques des pièces en conception mécanique est encore actuellement un problème important et non résolu de manière satisfaisante.

En effet, il n'existe pas de méthodes systématiques qui permettent de déterminer les limites admissibles de la géométrie des pièces fabriquées. Une méthode efficace doit être compatible avec les exigences du cahier des charges du système à assembler, c'est à dire correspondant à un tolérancement dit fonctionnel. À part quelques techniques basées sur la notion de chaînes de cotes, les tolérances dimensionnelles et géométriques indiquées au bureau d'étude font appel à l'expérience de l'entreprise.

En effet, les concepteurs doivent tracer les chaînes de cotes en identifiant tous les éléments fonctionnels dans un montage d'assemblage. Ils doivent ainsi identifier toutes les liaisons entre différentes pièces d'un montage. Cette méthode manuelle pour tracer des chaînes de cotes prend énormément de temps car il faut examiner minutieusement le schéma pour faire des chaînes de cotes optimales. Conscients de cette nécessité, nous nous proposons de déterminer des chaînes de cotes automatiquement.

### **2.4.2. L'approche préconisée**

La méthode consiste à établir une structure de données, à partir d'un dessin d'assemblage du produit en DAO, menant à un modèle de type graphe (appelé graphe des éléments fonctionnels). (Les éléments fonctionnels discutés ici sont en fait des points, des lignes et des surfaces, qu'ils soient fictifs ou réels.). Le concepteur identifie ensuite une paire d'éléments de ce graphe entre lesquels une condition jugée importante doit être respectée. Un algorithme d'identification de toutes les combinaisons possibles de paires d'éléments formant un cycle autour de la condition à respecter est ensuite exécuté.

L'algorithme proposé contient trois niveaux. Les premier et troisième niveaux sont plus étroitement liés aux moyens de définition géométrique (dessin DAO) et technologique des ensembles et des pièces mécaniques (affichage du résultat). Le deuxième niveau constitue pour sa part le noyau du traitement de la cotation fonctionnelle. Ce dernier contient trois modules. Son rôle est de générer de façon systématique le graphe de tous les éléments fonctionnels présents dans la représentation solide

de l'assemblage tel que tiré d'une base de données d'un logiciel DAO. Son deuxième rôle est l'identification des surfaces de contacts géométriques entre les différentes pièces. Pour cette fin, l'utilisateur sélectionne la condition fonctionnelle et les différentes paires de surfaces en contact chacune appartenant à une pièce différente. Après avoir élaboré le graphe des éléments fonctionnels, il s'agit enfin d'identifier toutes les combinaisons possibles de paires d'éléments formant un cycle autour de la condition à respecter telle que sélectionnée par l'utilisateur, et de remonter de paire en paire d'éléments fonctionnels jusqu'à un retour au point de départ. Cette étape de l'algorithme sert à établir l'arbre des éléments fonctionnels (AEF).

Le tableau 2.3 montre les étapes de l'algorithme

Tableau 2.3 : Les étapes de l'algorithme.

**Les trois modules du deuxième niveau de l'algorithme proposé :**

➤ **Établir la liste des éléments fonctionnels (LEF)**

- Lire la géométrie de chaque pièce de l'assemblage
- Établir tous les éléments fonctionnels dans chaque pièce

➤ **Établir le graphe des éléments fonctionnels (GEF)**

- À partir du (LEF) l'utilisateur sélectionne les éléments fonctionnels en contact (réel et/ou potentiel) entre deux pièces différentes ;
- Établir le graphe des éléments fonctionnels (GEF);

➤ **Élaborer l'arbre des éléments fonctionnels (AEF)**

- À partir du (LEF) l'utilisateur peut sélectionner la condition fonctionnelle à respecter dans le mécanisme en question ;
- À partir du (GEF) identifier les paires des éléments fonctionnels dans chaque pièce;
- À partir du (GEF) explorer le graphe de toutes les possibilités des paires des éléments fonctionnels qui ont un impact sur la condition fonctionnelle identifiée auparavant par l'utilisateur.
- Éliminer les redondances et les paires non convenables selon le concepteur;
- Identifier les dispersions pour chaque paire d'éléments fonctionnels.

Les chapitres suivants discutent en détail des différents modules du deuxième niveau de l'algorithme.

## CHAPITRE 3

### ALGORITHME D'IDENTIFICATION DES CHÂÎNES DE COTES

---

#### 3.1 ÉLÉMENTS DE BASE DE L'IDENTIFICATION DES CHÂÎNES DE COTES

- 3.1.1. Éléments fonctionnels (EFs)
  - 3.1.1.1. Définition
  - 3.1.1.2. Implantation
- 3.1.2. Surfaces en contact
  - 3.1.2.1. Définition
  - 3.1.2.2. Identification des surfaces en contact
- 3.1.3. Condition fonctionnelle (CF)
  - 3.1.3.1. Définition
  - 3.1.3.2. Identification d'une Condition fonctionnelle
- 3.1.4. Paires des éléments fonctionnels (PEFs)
  - 3.1.4.1. Définition
  - 3.1.4.2. Combinaisons des tolérances possibles

#### 3.2 GRAPHE D'ÉLÉMENTS FONCTIONNELS (GEF)

- 3.2.1. Définition d'un graphe des éléments fonctionnels
- 3.2.2. Implantation d'un GEF
  - 3.2.2.1. Représentation d'un GEF
  - 3.2.2.2. Exploitation de la théorie de graphe
  - 3.2.2.3. Algorithme d'implantation du GEF
- 3.2.3. Identification des paires des éléments fonctionnels
- 3.2.4. Affectation de tolérance sur une paire d'éléments fonctionnels
- 3.2.5. Existence et dénombrement des chemins dans un graphe

#### 3.3 GÉNÉRATION DE L'ARBRE DES ÉLÉMENTS FONCTIONNELS (AEFS)

- 3.3.1. Définition
- 3.3.2. Représentation graphique
- 3.3.3. Implantation



Ce chapitre a pour objectif l'élaboration d'un algorithme servant à l'identification des chaînes de cotes.

Pour réaliser cet algorithme on commence par établir un modèle mathématique pour le graphe des éléments fonctionnels en se référant à la théorie des graphes. Une fois le modèle graphique mis au point on pourra énoncer l'algorithme ayant pour tâche l'exploration du graphe afin de détecter toute chaîne de cotes tridimensionnelle.

Pour obtenir le modèle et l'algorithme nous allons en premier définir les différents termes de base pour celui-ci.

### 3.1 ÉLÉMENTS DE BASE DE L'IDENTIFICATION DES CHÂÎNES DE COTES

Pour illustrer l'algorithme nous allons utiliser l'exemple de contre pointe, figure 3.1.

Considérons le montage suivant qui contient deux conditions fonctionnelles. Le mécanisme de contre-pointe fixe est composé de trois pièces. La pièce 3 est bloquée par l'équerre 2 qui reçoit la pointe 1 dans son alésage.

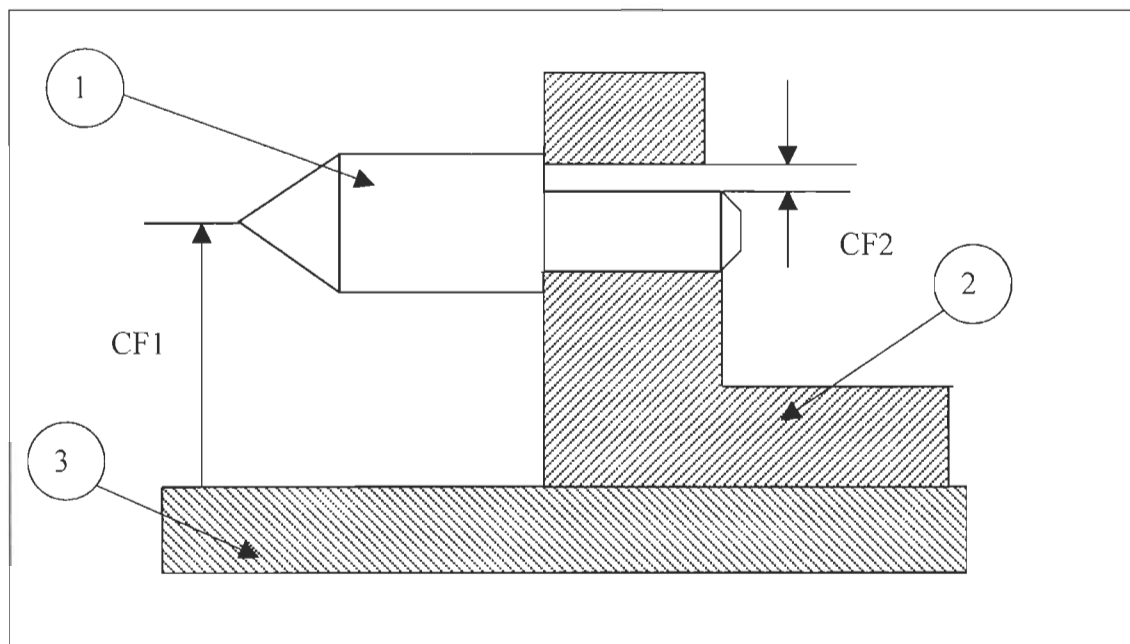


Figure 3.1 : Mécanisme tenant deux conditions fonctionnelles

**Note** : Cet exemple est utilisé tout au long de ce document.

### 3.1.1. Éléments fonctionnels (EFs)

#### 3.1.1.1. Définition

Dans un mécanisme nous avons toujours deux ou plusieurs pièces. Chaque pièce est un assemblage de surfaces élémentaires auxquelles on peut associer des éléments géométriques connus. Dans la plupart des assemblages mécaniques les éléments géométriques utilisés sont : le point, la droite, le plan, le cercle, la sphère, le cylindre ou le cône. Dans les autres cas, les éléments géométriques utilisés peuvent être définis par une succession de points et de normales en ces points (B-Spline). Ces éléments géométriques sont les éléments fonctionnels dans les pièces à assembler, ces éléments peuvent être réels (Contour d'un cylindre, Cercle,...) ou fictifs (point, axe d'un cylindre, rencontre des deux plans « courbe »,...). (Figure 3.2). L'ensemble de tous ces éléments EFs est noté par L (Liste des éléments fonctionnels).

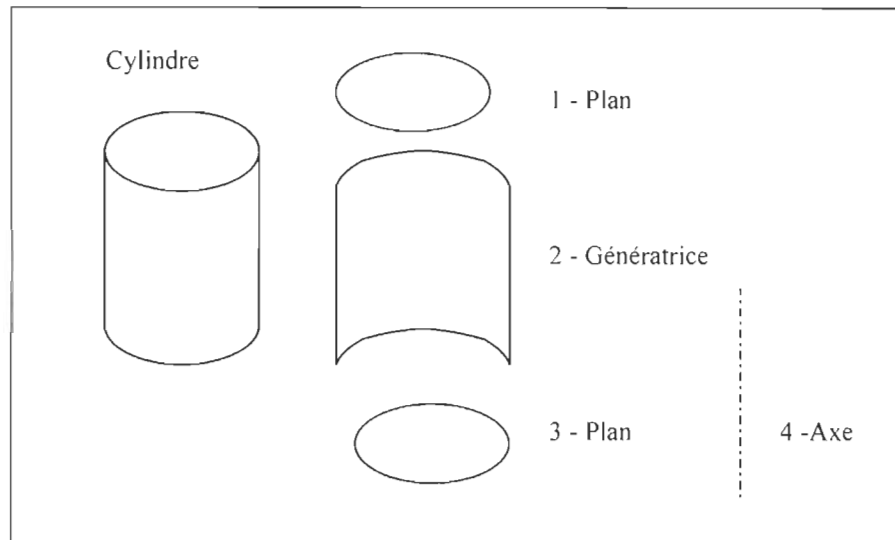


Figure 3.2 : Éléments fonctionnels d'un cylindre

La notation d'un EF utilisé dans notre travail est un crochet (figure 3.3 ) qui contient deux chiffres séparés par une virgule pour identifier le numéro de la pièce et celui de l'élément fonctionnel :

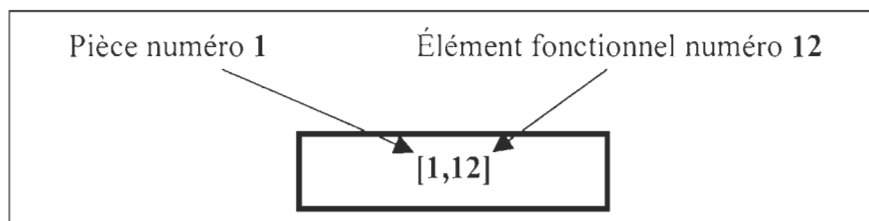


Figure 3.3 : Exemple d'un élément fonctionnel

Afin d'obtenir ces éléments fonctionnels nous appliquerons une approche qui consiste à utiliser les informations contenues dans le fichier texte BREP «.sat» (Boundary Représentation). Ce fichier s'obtient automatiquement après la modélisation solide d'un mécanisme quelconque. Dans ce fichier, les contours, les arêtes, les sommets et la géométrie des surfaces de chaque face, sont décrites par la combinaison de formats spéciaux.

### 3.1.1.2 Implantation

Les modèles BREP (Boundary Representation ) contiennent des informations géométriques (coordonnées de points, équations de faces ...) et topologiques (sur les arêtes et les orientations). Ce fichier contient tous les éléments fonctionnels d'un mécanisme (Surfaces de contour et courbe de rencontre etc.) la figure 3.4 montre un exemple d'un cylindre.

Cette représentation géométrique des frontières est décrite par des entités géométriques: body, lump, shell, face, loop, coedge, edge, vertex, transform. Dans ce fichier, les champs contenus dans les lignes commençant par ces entités ont dans chacun une signification. Dans chaque champ on trouve des pointeurs qui est le numéro des lignes concernant. Exemple : la figure montre les détails d'un cylindre ce fichier contient un seul solide alors on se trouve avec un seul (body) et dans la ligne body il y aura un pointeur pour le lump et dans celui un pointeur pour le shell correspondant qui est la ligne 2 etc. pour trouver l'équation de la surface qui est ici un cône avec deux diamètres majeur et mineur égaux. (Pour plus de détail consulter le manuel Acis développer spécialement pour ce fichier).

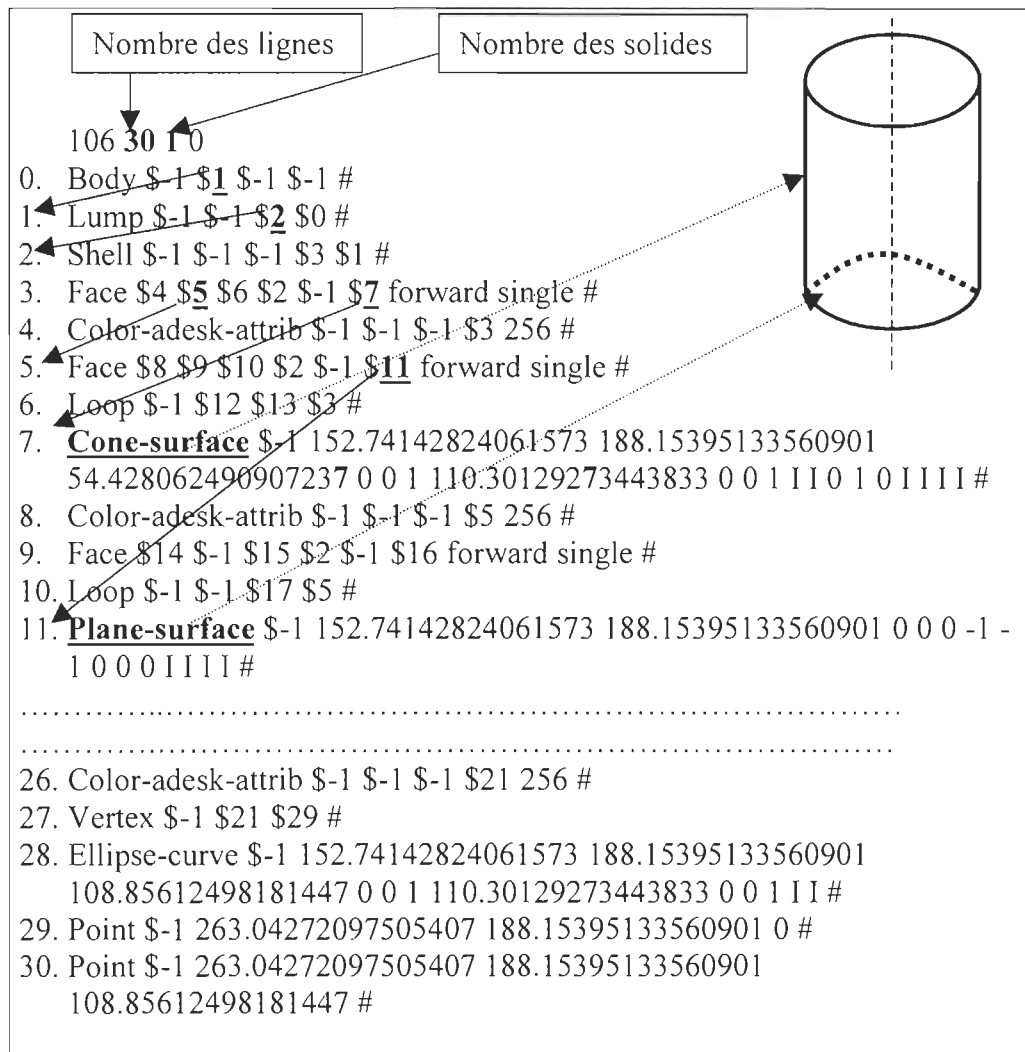


Figure 3.4 : Exemple d'un fichier BREP

Ce fichier est bien structurer, il faut tout simplement lire le fichier et le mémoriser dans un tableau à une dimension pour retracer le pointeur de chaque ligne.

Afin de réaliser la liste des éléments fonctionnels **L** nous avons implanté une structure de données qui est une liste chaînée à deux dimensions. La liste chaînée contient deux enregistrements, l'un pour les EFs et l'autre pour l'élément suivant. On peut imaginer cette liste comme un tableau, la première colonne est le nombre de pièces et les lignes sont les EFs , le lien entre les différents éléments est fait par un pointeur afin de faciliter la gestion dynamique de la mémoire (l'utilisation de la commande NEW et DELETE est importante). Cette structure est présentée graphiquement à la (figure 3.5). L'algorithme servant à la lecture du fichier BREP est illustré comme suit (tableau 3.1)

**LireBrep** ( char \*NomFichier , Liste L[ ] ) est une fonction qui consiste premièrement à déterminer le nombre de pièces dans le mécanisme et deuxièmement à établir tous les EFs de chaque pièce pour obtenir la liste des éléments fonctionnels L ( tableau 3.2 ).

Tableau 3.1 : Algorithme de lecture du fichier BREP

```

int CACis::LireBrep(char *NomFichier,Liste L[] )
{
    OUVRIR le fichier BREP (NomFichier)
    Lire le nombre des lignes dans le fichier BREP
    Lire le nombre des solides dans le fichier BREP
    Lire et mémoriser le fichier BREP dans un tableau

    POUR chaque Solide
        Lire informations face
            Tant que le solide contient des faces
                Lire les attributs de faces
                Lire l'équation face
                Mémoriser les attributs et les équations dans la liste L
            Fin tant que

        Lire informations courbes
            Tant que le solide contient des courbes
                Lire les attributs de courbe
                Lire l'équation courbes
                Mémoriser les attributs et les équations dans la liste L
            Fin tant que

    Fin pour
    FERMER le fichier BREP (NomFichier).
}

```

**NOTE** : CACis est une classe développée en C++, son rôle est la lecture et la mémorisation des éléments fonctionnels dans le fichier BREP qui contient les géométries du mécanisme en question.

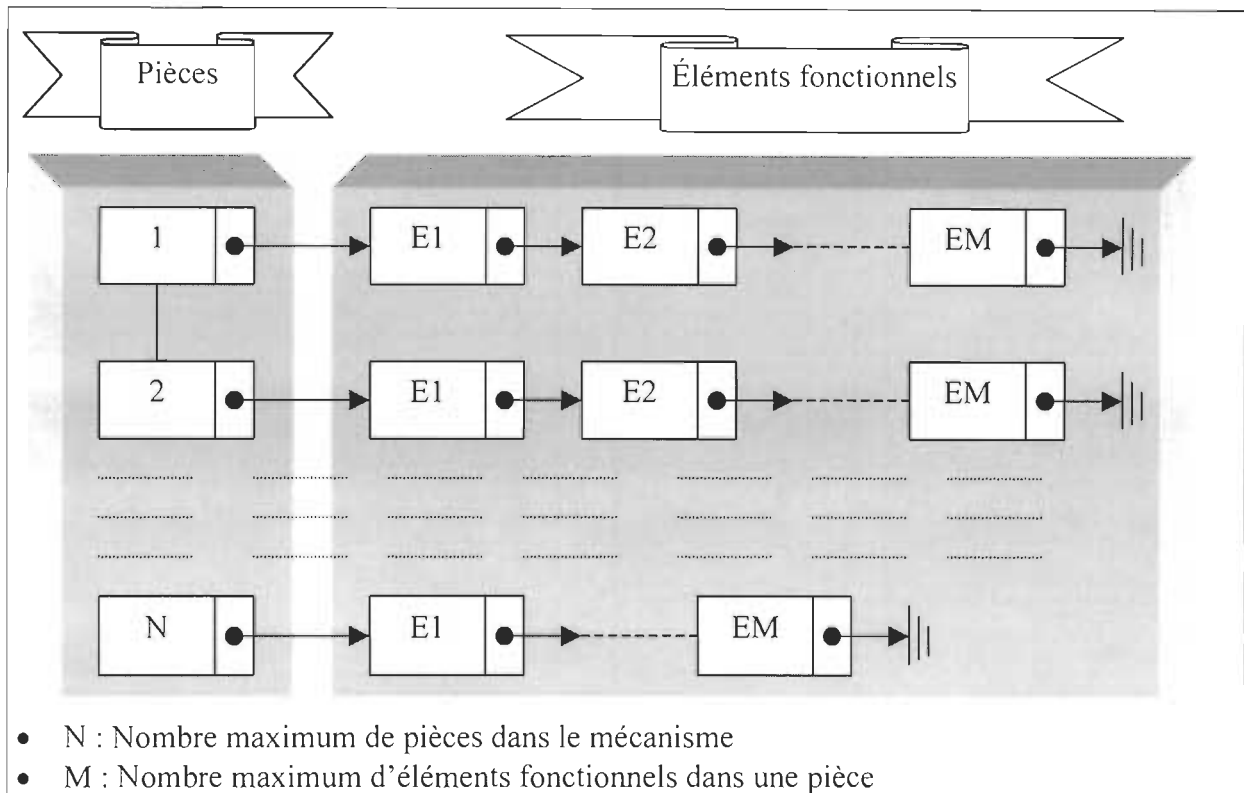


Figure 3.5 : Représentation de la liste des éléments fonctionnels en mémoire

Pour notre exemple nous obtenons trois pièces (figure 3.6). La liste des EFs est la suivante (tableau 3.2)  $L = \{ [1,1], [1,2], [1,3], [1,4], [1,5], [1,6], [1,7], [1,8], [2,1], [2,2], [2,3], [2,4], [2,5], [2,6], [2,7], [2,8], [2,9], [3,1], [3,2], [3,3], [3,4], [3,5], [3,6] \}$ .

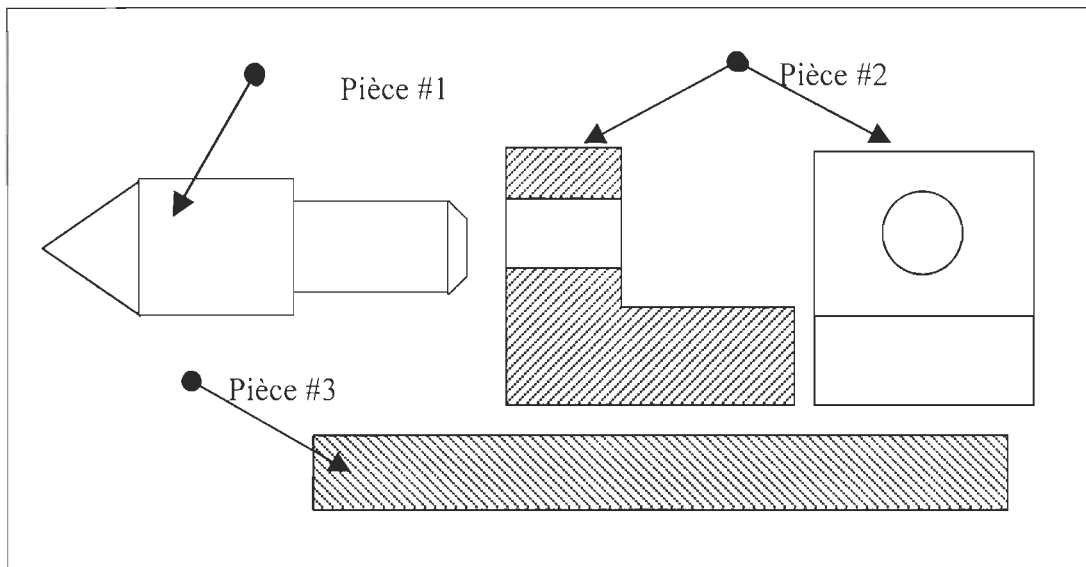
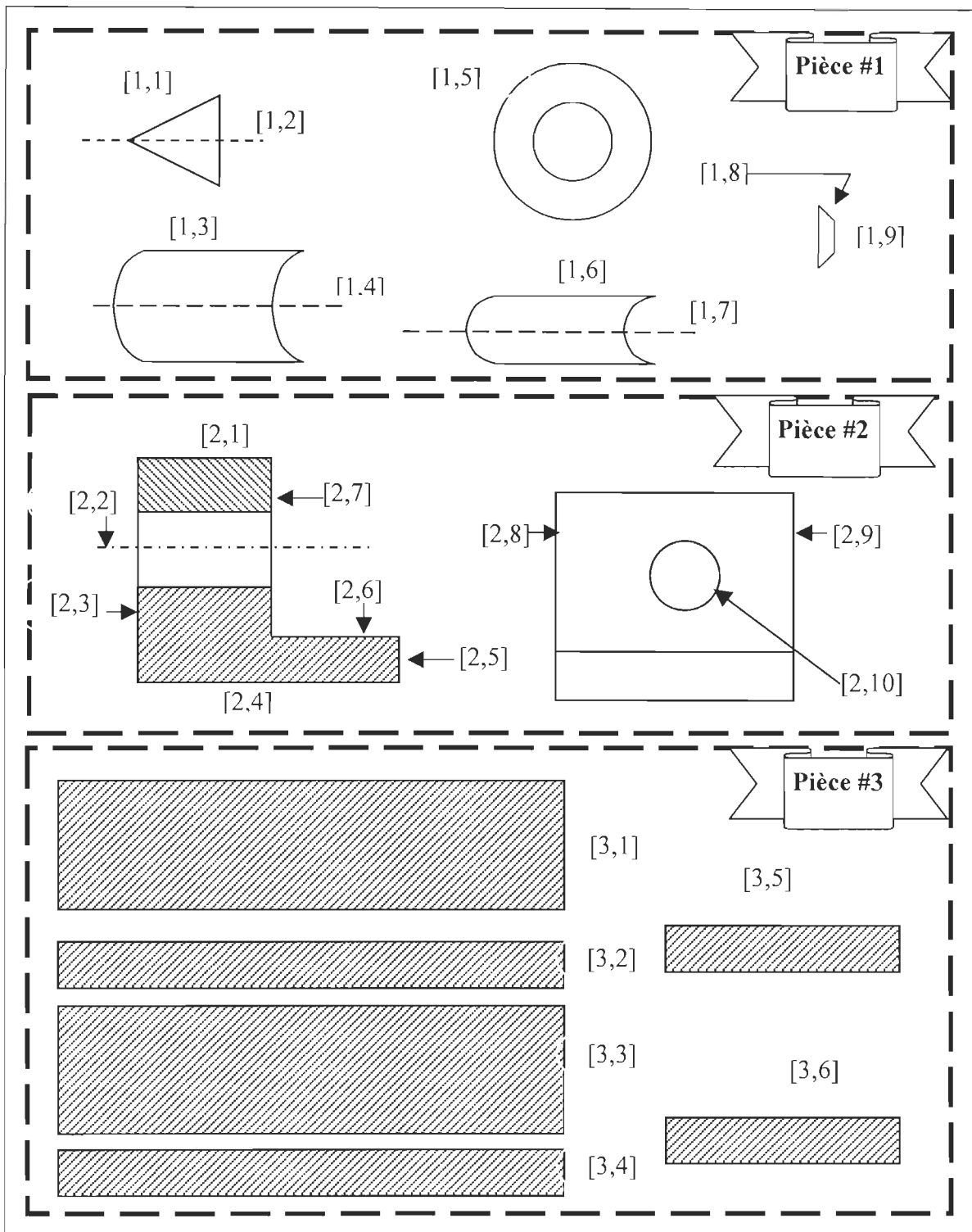


Figure 3.6 : Différentes pièces dans le montage de contre pointe

Le tableau suivant nous montre tous les éléments fonctionnels du mécanisme contre-pointe

Tableau 3.2 : Liste des éléments fonctionnels



Nous pouvons remarquer que ce tableau contient tous les éléments fonctionnels existants dans le mécanisme pour ensuite déterminer les éléments des contacts pour obtenir le graphe d'éléments décrit ultérieurement. Le graphe à son tour ne contient que d'éléments fonctionnels en contact et une condition fonctionnelle.

### 3.1.2. Surfaces en contact

#### 3.1.2.1 Définition

Deux éléments fonctionnels ont une relation de contact s'il existe un contact physique entre ces deux éléments fonctionnels sur des pièces différentes dans un mécanisme quelconque à assembler. La relation de contact est une relation binaire symétrique sur  $L$ . Par conséquent, les composants d'incidence dans une relation de contact sont nécessairement des éléments de  $L \times L$  où  $\times$  dénote le produit cartésien sur  $L$ . L'ensemble de tous les contacts de celui-ci est dénoté par  $C$ .

La figure (3.7) montre un exemple d'éléments en contact.

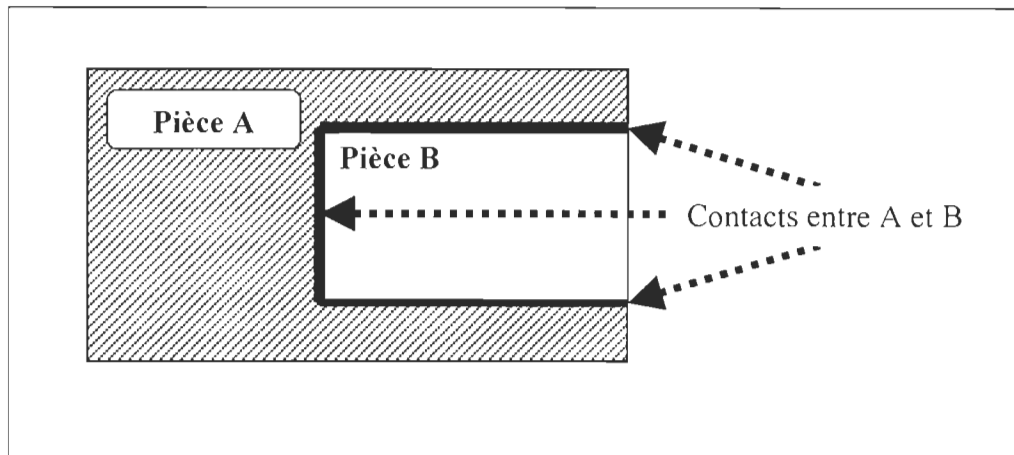


Figure 3.7 : Exemple de composants en contact

Dans cet exemple, on peut identifier trois éléments fonctionnels en contacts entre les deux pièces A et B. Ils sont des contacts plan-plan.

#### 3.1.2.2 Identification des surfaces en contact

L'identification des contacts entre les pièces est un problème à traiter afin de trouver le graphe des éléments fonctionnels. Il y a deux possibilités pour l'identification de ces éléments en contact. La première consiste à effectuer des tests mathématiques sur tous les contacts possibles entre les différents éléments de chaque pièce, cette méthode était élaborée par Bre [14, 15]. Cependant cette méthode est assez longue en développement et risque de ne pas prendre en considération toutes les possibilités. La deuxième possibilité consiste à laisser l'utilisateur sélectionner les surfaces en contacts. Nous avons opté pour la deuxième méthode afin de valider notre algorithme.

On utilise la liste des éléments de l'exemple de la contre pointe obtenue auparavant pour déterminer manuellement les éléments en contact.

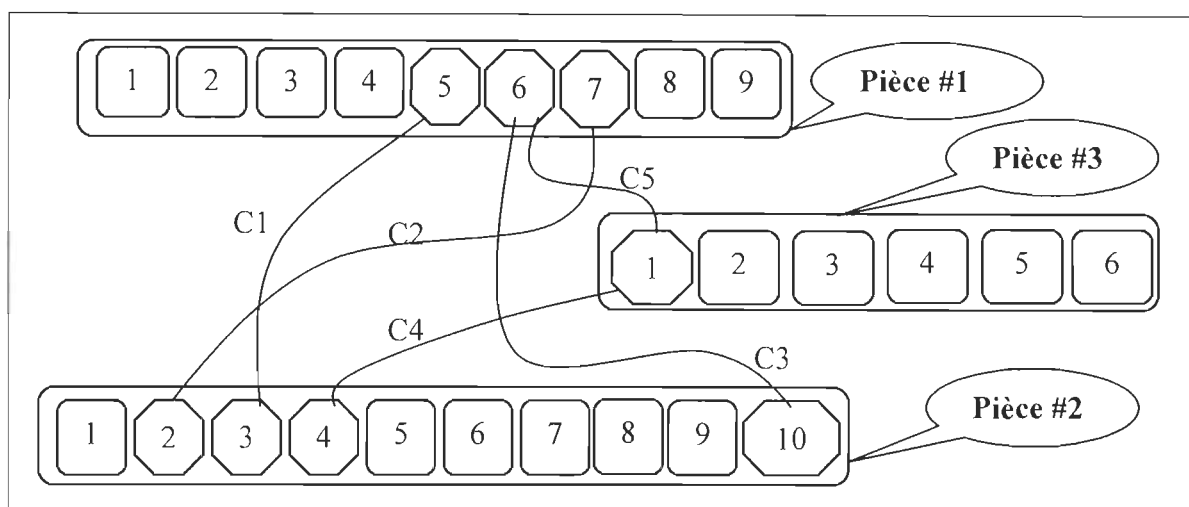


Figure 3.8 : Sélection des contacts des éléments fonctionnels

L'exemple contient cinq contacts :

$$C1 = ([1,5], [2,3]) \quad C2 = ([1,7], [2,2]) \quad C3 = ([1,6], [2,10]) , \\ C4 = ([2,4], [3,1]) \text{ et } C5 = ([3,1], [1,6]).$$

Après les sélections de contacts entre ces différents éléments on obtient l'ensemble des contacts C suivant:

$$C = \{ C1 , C2 , C3 , C4 , C5 \}$$

### 3.1.3 Condition fonctionnelle (CF)

#### 3.1.3.1 Définition

Une condition fonctionnelle nécessaire à l'assemblage représente l'espace compris entre deux surfaces terminales, c'est-à-dire des surfaces qui ne sont pas sur une même pièce. Sur un dessin, on symbolise la condition fonctionnelle par une flèche double orientée. La figure 3.9 illustre le principe d'une CF. La condition fonctionnelle est une relation binaire non symétrique sur L, contrairement à l'ensemble C. Deux éléments fonctionnels d'incidence forment une condition fonctionnelle appartenant nécessairement à  $L \times L$ . L'ensemble disjoint d'une condition fonctionnelle est dénoté par J.



Le montage suivant illustre bien le principe d'une condition fonctionnelle :

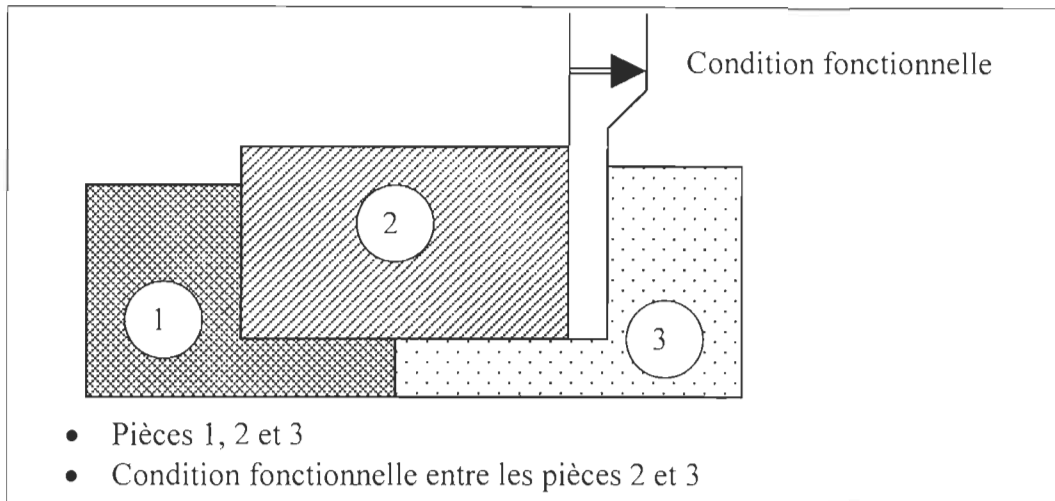


Figure 3.9 : Condition fonctionnelle

### 3.1.3.2 Identification d'une Condition fonctionnelle

Afin d'identifier une condition fonctionnelle, nous avons utilisé le même principe que pour des surfaces de contacts. La figure 3.10 montre la sélection d'une condition fonctionnelle. Nous avons une condition fonctionnelle à respecter entre l'élément [3,1] et [1,2].

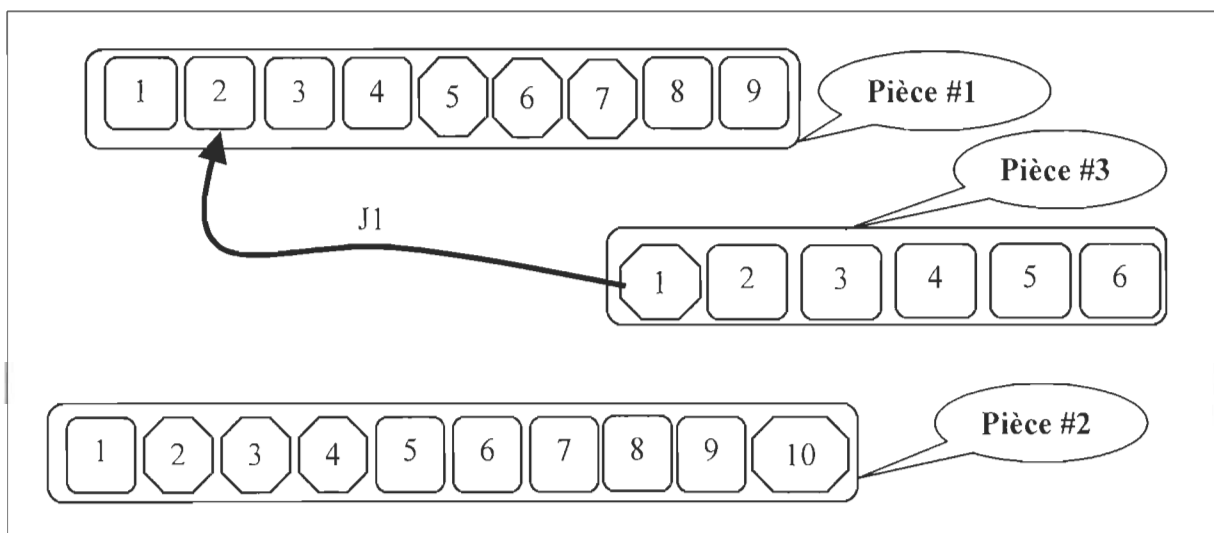


Figure 3.10 : Sélection d'une condition fonctionnelle

Après la sélection d'une condition fonctionnelle entre les deux éléments fonctionnels sur les deux pièces différentes on obtient l'élément  $J1 = ([3,1], [1,2])$ . L'ensemble disjoint  $J$  devient :

$$J = \{J1\}$$

### 3.1.4 Paires des éléments fonctionnels (PEFs)

#### 3.1.4.1 Définition

Deux éléments fonctionnels forment une paire d'éléments fonctionnels si les deux éléments sont dans une même pièce et les deux éléments ont des contacts avec d'autres pièces. La notation d'une PEF que nous avons adoptée est une parenthèse qui contient deux éléments fonctionnels séparés par deux points. (Figure 3.11). L'obtention des ces paires sera détailler ultérieurement dans la section 3.2.3.

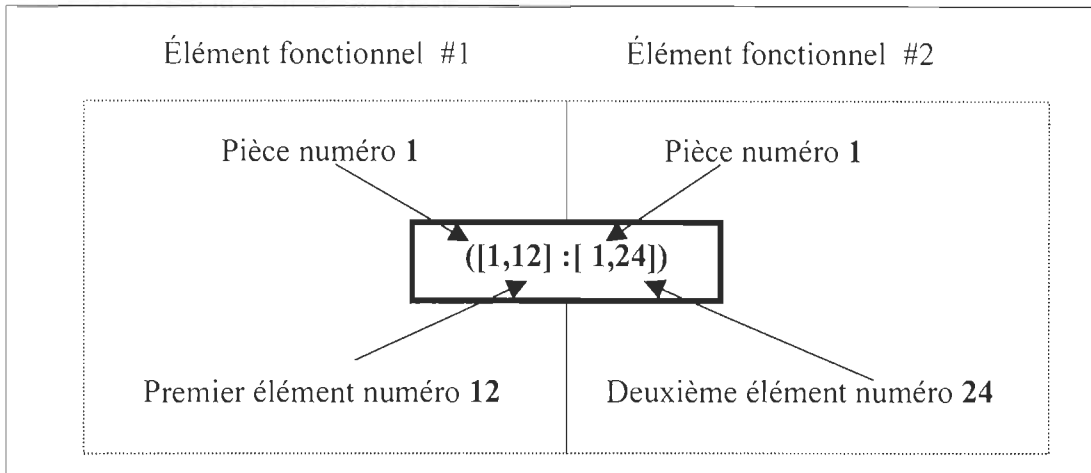


Figure 3.11 : Exemple d'une notation d'une paire d'éléments fonctionnels

1. Le premier chiffre est réservé pour le numéro de la pièce;
2. Le deuxième pour identifier l'élément fonctionnel de départ;
3. Le troisième devrait être le même que le premier qui est le numéro de la pièce;
4. Le quatrième sert à identifier l'élément fonctionnel d'arrivée.

À l'exemple suivant (figure 3.12), on peut identifier deux paires d'éléments fonctionnels dans chacune des pièces A et B.

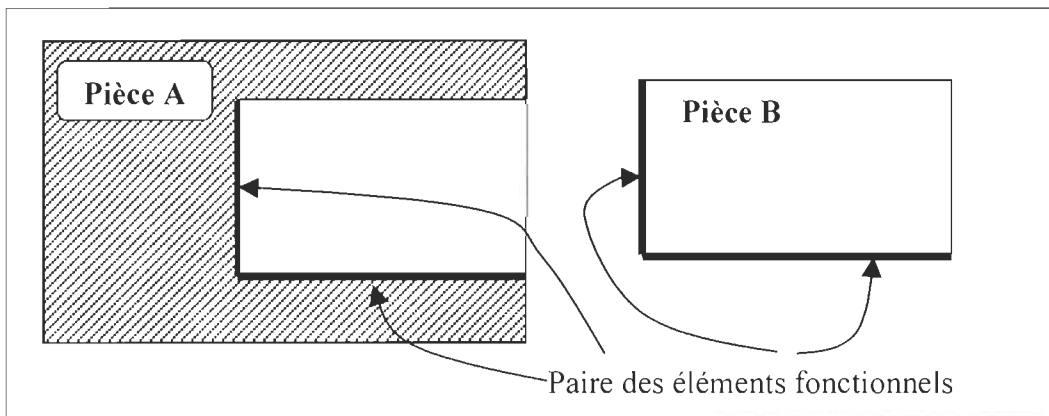


Figure 3.12 : Exemple d'une paire d'éléments fonctionnels










Un cas particulier : un élément peut former avec lui-même une paire seulement si l'élément est le seul à faire contact avec d'autres éléments d'une autre pièce. La détermination des paires d'éléments dans une pièce sera discutée dans la section 3.2.3.

### 3.1.4.2 Combinaisons des tolérances possibles

Trois types de défaut peuvent être énumérés pour des paires d'éléments fonctionnels. Ces trois types sont les défauts d'orientation, les défauts de position et ceux de battement. Ces types de défauts s'appliquent simplement au cas où il existe deux éléments à tolérer. C'est le cas de notre exemple : nous avons des paires d'éléments fonctionnels et ces paires appartiennent à une même pièce.

Le tableau 3.3 montre les trois différents type de défauts qu'on peut rencontrer en cotation standard, avec l'association des différents éléments fonctionnels. (Ce tableau est un résumé de celui de 2.2).

Tableau 3.3 : Combinaisons des tolérances pour des paires d'éléments fonctionnels

Type défaut	Paires d'éléments fonctionnels	Symbole
Orientation	Parallélisme entre lignes et surfaces	
	Perpendicularité entre droites et surfaces	
	Inclinaisons entre droites et surfaces	
Position	Localisation d'un point, d'une ligne ou d'une surface.	
	Concentricité entre points	
	Concentricité entre droites	
	Symétrie de lignes ou surfaces	
Battement	Battement simple d'une ligne	
	Battement total d'une surface	

Pour déterminer le type de tolérances à appliquer, il faut tout simplement connaître la géométrie des éléments qui forment une paire d'éléments fonctionnels.

## 3.2 GRAPHE D'ÉLÉMENTS FONCTIONNELS (GEF)

Avant de nous intéresser à la génération de l'arbre des éléments fonctionnels, nous nous préoccupons d'abord de la représentation du mécanisme. Nous présenterons un modèle de graphe qui nous servira, par la suite, pour la génération de l'arbre des paires d'éléments fonctionnels.

### 3.2.1. Définition d'un graphe des éléments fonctionnels

Deux éléments fonctionnels ont une relation entre eux s'ils ont une relation de contact ou une condition fonctionnelle. L'ensemble des relations est stocké dans une liste de relations fonctionnelles. Cette liste est notée par  $R$  qui satisfait l'équation suivante :

$$R = C \cup J \quad (3.1)$$

$\psi()$  est une fonction d'incidence qui associe chaque relation à ses composants d'incidence. Le domaine de cette fonction est  $R$  et l'image est élément de  $L \times L$ .

$$\psi(r) = (\text{Plan}, \text{Axe\_pointe})$$

Où  $r$  est une condition fonctionnelle entre les deux éléments Plan et Axe\_Pointe.

À partir de ces définitions, le graphe des éléments fonctionnels peut être défini par la relation simple  $G = \{L_G, R_G, \psi_G\}$ . La figure 3.13 montre le modèle graphique de l'exemple précédent.

La construction de ce graphe est relativement simple. Les pièces du mécanisme constituent les nœuds ou sommets du graphe tandis que les relations (contacts ou conditions fonctionnelles) correspondent aux arcs ou liaisons de ce même graphe. La différence entre ce type de graphe et le graphe conventionnel est que les éléments fonctionnels sont distingués les uns des autres à l'intérieur d'un sommet ou d'une pièce.

Le graphe peut se décrire comme suit :

- Les grands cercles (sommets) correspondent aux pièces;
- Les rectangles arrondis aux éléments fonctionnels;
- Les liaisons (ou arcs) aux contacts réels ou potentiels dans le mécanisme.

On peut remarquer que chaque élément fonctionnel peut recevoir plus d'un contact provenant d'une ou plusieurs pièces. Toutefois, il n'est pas possible d'avoir plus d'un contact entre les deux mêmes éléments. En effet, physiquement, une telle situation n'aurait pas de sens.

On peut remarquer aussi que seulement les éléments fonctionnels qui ont des contacts avec d'autres éléments d'une pièce différente sont pris en considération parce que ces éléments peuvent influencer les chaînes de cotes.

La représentation suivante est celle du GEF initial de l'exemple de la contre pointe (Figure 3.1)

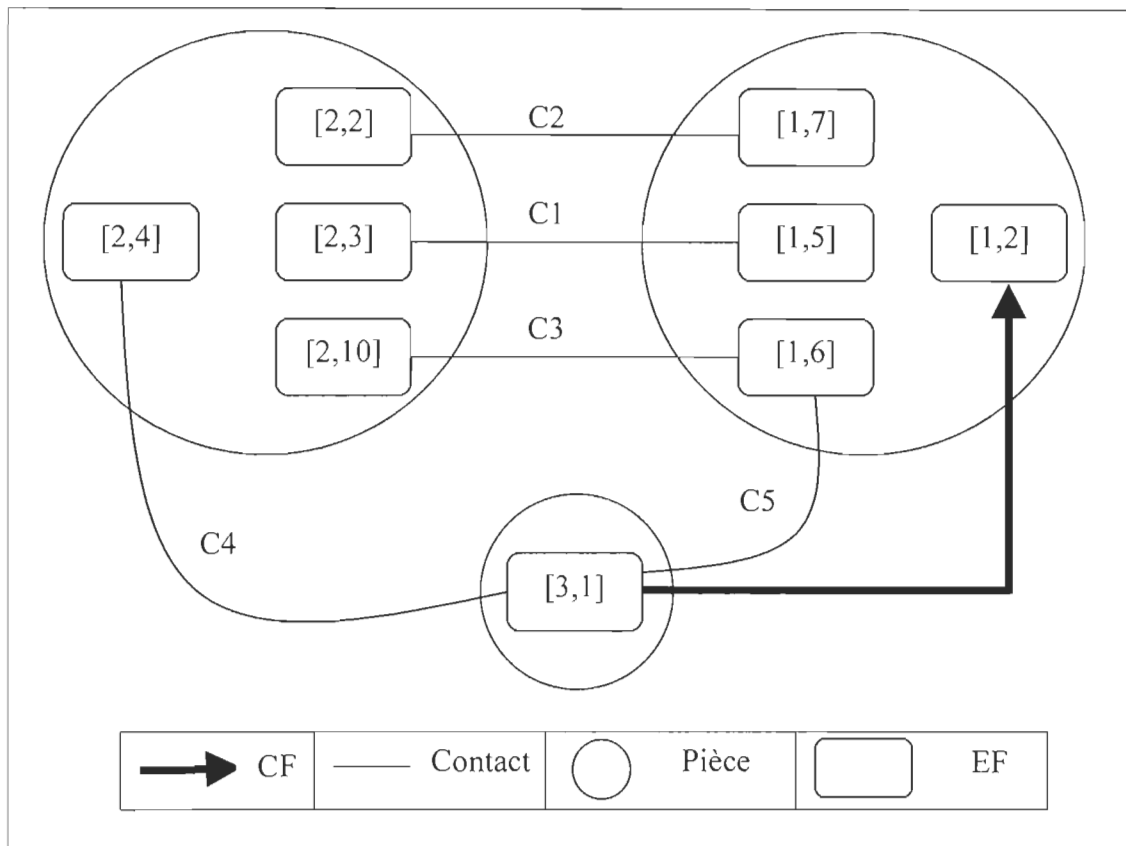


Figure 3.13 : Graphe des éléments fonctionnels initial

Nous pouvons remarquer que, le GEF (figure 3.13) montre seulement les contacts entre différentes pièces les paires d'éléments n'apparaissent pas. La connexion entre les différents éléments dans une même pièce est obtenue en fonction de la condition fonctionnelle. Cette connexion donne les différentes paires d'éléments fonctionnels (ce principe est illustré plus loin, pour plus de détails techniques voir le chapitre 5).

Un graphe G est connexe (connected) s'il existe une chaîne entre toute paire de sommet. Le graphe est donc «un seul bloc ». Si G n'est pas connexe on peut identifier plusieurs sous-graphes connexes. (figure 3.14)[16, 17, 18]

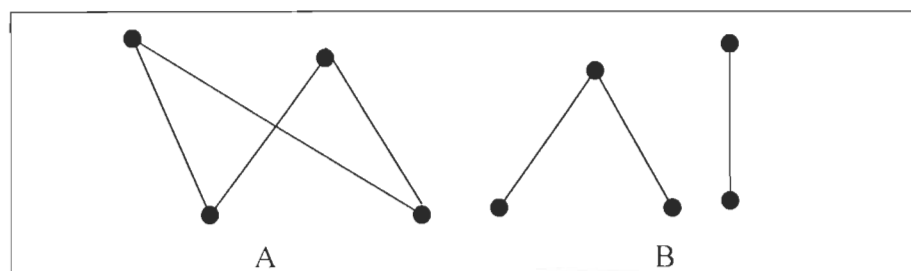


Figure 3.14 : Types de graphe A) Connexe B) Non connexe

En tolérancement dans un mécanisme les pièces sont reliées entre elles par une série de contacts. Dans le cas contraire, cela signifierait que certaines pièces n'ont aucun lien, que ce soit direct ou indirect, avec d'autres pièces de l'assemblage; nous n'aurions pas un mécanisme complètement assemblé dans ce cas. On peut conclure que nous avons un graphe connexe non orienté.

Dans un graphe d'éléments fonctionnels tous les éléments dans une pièce sont connectés par des paires d'éléments entre eux parce que tous les éléments dans le GEF ont au moins un contact.

L'élément [3,1] est le départ de la condition fonctionnelle qu'il devrait être la tête du GEF, nous pouvons déterminer à partir de cet élément les éléments qui peuvent former contact avec celui-ci. Un seul élément peut avoir un contact avec [3,1] celui [2,4]. Et l'élément [2,4] peut former paire avec { [2,2], [2,3], [2,10] } c'est à dire l'élément [2,4] peut former paire d'éléments avec tous les éléments de la même pièce sauf lui-même. Ensuite l'élément [2,2] fait contact avec [1,7]. Pour que l'élément [1,7] forme paire avec [1,2] (Figure 3.15)

La représentation suivante est celle du GEF pour les éléments [3,1], [2,4], [2,2] et [1,7] de l'exemple de la contre pointe.

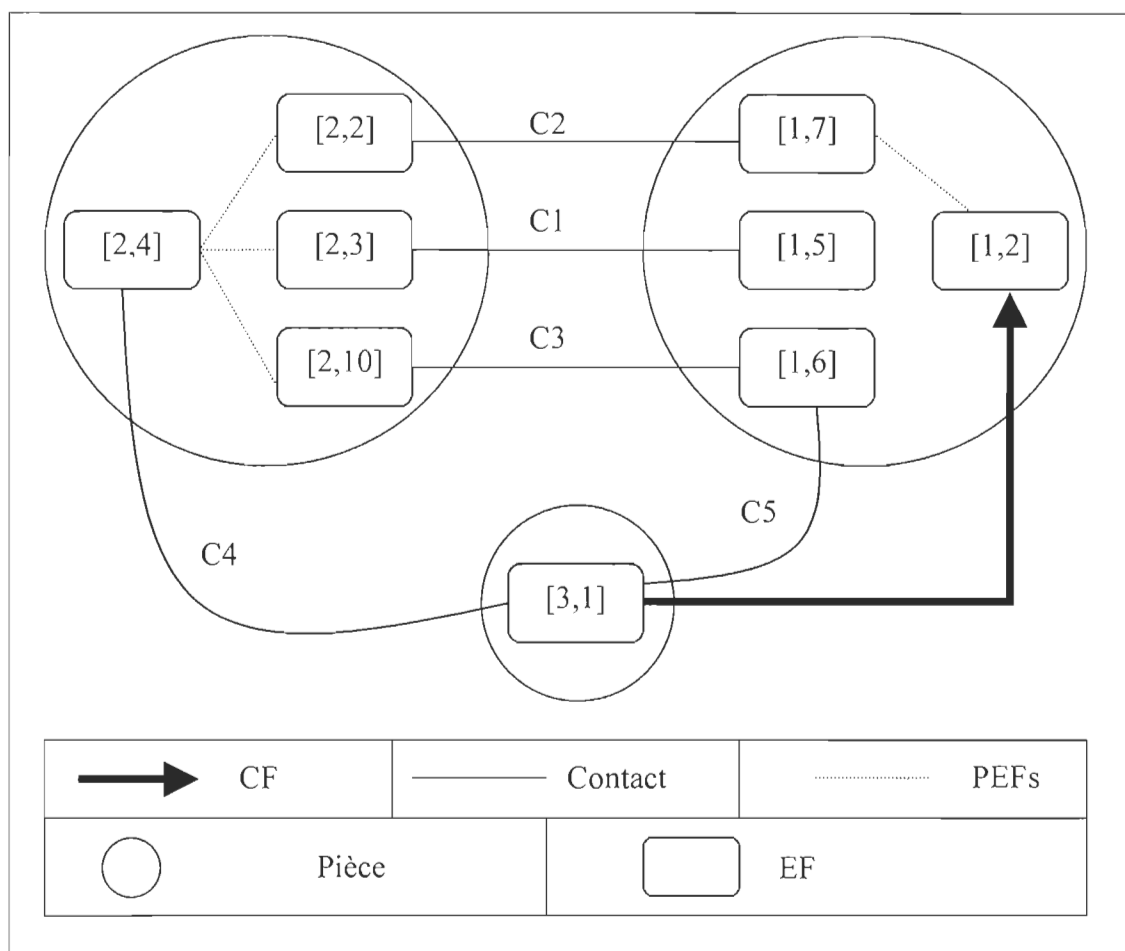


Figure 3.15 : Graphe des éléments fonctionnels intermédiaire

Nous pouvons ensuite suivre le processus pour trouver le GEF complet.

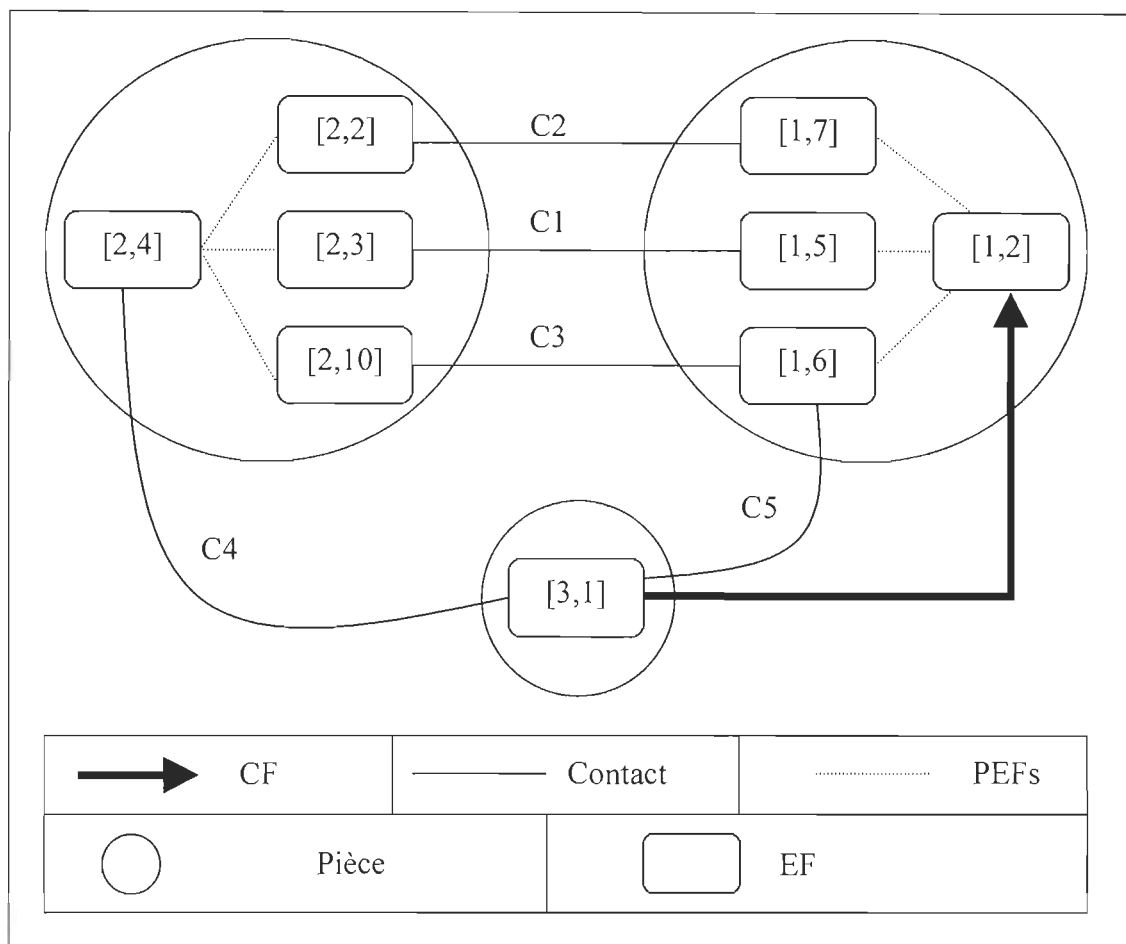


Figure 3.16 : Graphe des éléments fonctionnels complet

Remarque :

Un élément appartient à la dernière pièce ne peut former paire d'éléments qu'avec l'élément d'arrivée de la condition fonctionnelle.

On peut présenter le graphe précédent par les éléments fonctionnels (figure 3.17) mais cette construction cause une difficulté pour la réalisation de l'arbre de chaînes des cotes plus tard dans ce travail. En d'autres termes le problème est de déterminer le cycle autour d'une condition fonctionnelle.

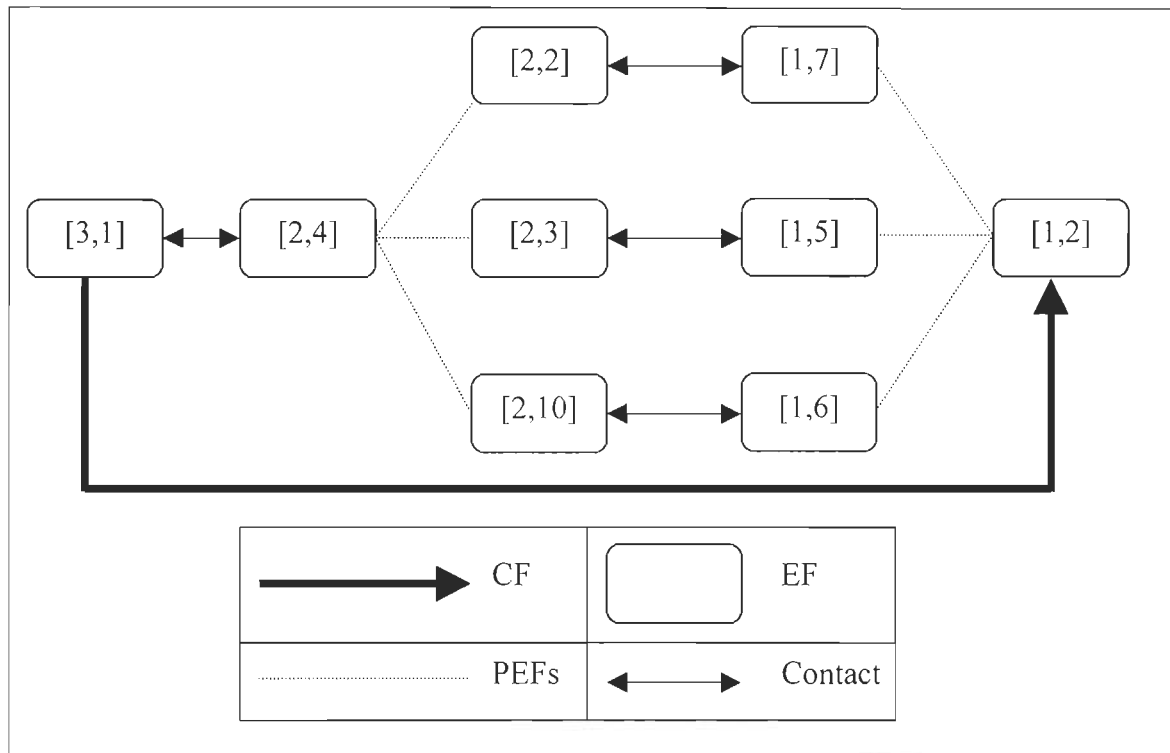


Figure 3.17 : Graphe des éléments fonctionnels «groupement par éléments fonctionnels »

Dans cette représentation il n'est pas possible d'identifier les paires d'éléments fonctionnels qui devraient appartenir à une même pièce parce que tous les éléments sont présentés ici dans un seul graphe. Par contre la première représentation a l'avantage de placer les éléments des différentes pièces dans des sous-graphes différents et ceux-ci dans un graphe global.

Dans une chaîne on ne peut pas trouver une pièce plus d'une fois, dans ce cas un problème peut se poser plus tard quand on veut traverser le graphe pour obtenir l'arbre d'éléments fonctionnels. Le parcours du graphe figure 3.17, il est difficile de vérifier si une pièce se trouve plus qu'une fois dans une chaîne.



### 3.2.2 Implantation du GEF

#### 3.2.2.1 Représentation d'un GEF

Afin de traiter un GEF à l'aide d'une machine (Ordinateur), il est nécessaire de concevoir une représentation informatique. Nous étudierons deux représentations couramment utilisées ; on choisit habituellement l'une ou l'autre selon que le graphe est dense (trop de connexion) ou creux (moins de connexion) ; mais, comme toujours, les opérations devant être réalisées jouent aussi un rôle déterminant.

Soit le graphe simple non orienté :  $G = \{ L_G, R_G, \psi_G \}$ .

Ensemble des éléments (sommets) :  $L_G = \{1,2,3,4,5,6,7\}$

Ensemble de relation (arêtes) :

$$R_G = \{ [1,1], [1,2], [2,3], [2,4], [3,4], [3,5], [4,6], [6,5], [5,7], [6,7], [2,1], [3,2], [4,2], [4,3], [5,3], [6,4], [5,6], [7,5], [7,6] \}$$

La fonction d'incidence :

$$\begin{aligned} \psi_G(E1) &= \{1, 1\}, \psi_G(E2) = \{1, 2\}, \psi_G(E3) = \{2, 3\}, \\ \psi_G(E4) &= \{2, 4\}, \psi_G(E5) = \{3, 4\}, \psi_G(E6) = \{3, 5\}, \\ \psi_G(E7) &= \{4, 6\}, \psi_G(E8) = \{5, 6\}, \psi_G(E9) = \{5, 7\}, \\ \psi_G(E10) &= \{6, 7\}. \end{aligned}$$

#### La première représentation :

La représentation la plus directe des graphes est celle de la *matrice d'incidence* (matrice cartésienne) et celle de la matrice d'adjacence.

**Définition de la matrice d'incidence** (Figure 3.18):

La matrice d'incidence  $A$  sommets-sommets de  $G$  est une matrice à coefficients 0 ou 1, où chaque ligne  $i$  correspond à un sommet  $i$  de  $G$ , et chaque colonne à une arête  $R_i = (i, j)$  de  $G$  ( $R_i \in R_G$ ).

**Définition de la matrice d'adjacence** (Figure 3.18):

La matrice d'adjacence (ou : matrice d'incidence sommets-sommets) est une matrice à coefficients 0 ou 1 :

$$B = \sum_{i=1}^N \sum_{j=1}^N B_{(i,j)}$$

Où chaque ligne correspond à un sommet de  $G$ , où chaque colonne correspond à un sommet de  $G$ , et où

$$B_{(i,j)} = 1 \text{ si et seulement si } (i, j) \in L_G \text{ (} B_{(i,j)} = 0 \text{ sinon)}$$

Dans le cas d'un graphe non orienté, on peut aussi définir la matrice d'adjacence d'un graphe simple en considérant qu'à chaque arête ( i , j ), correspondent deux arcs ( i , j ) et ( j , i). Dans ce cas la matrice d'adjacence est symétrique.

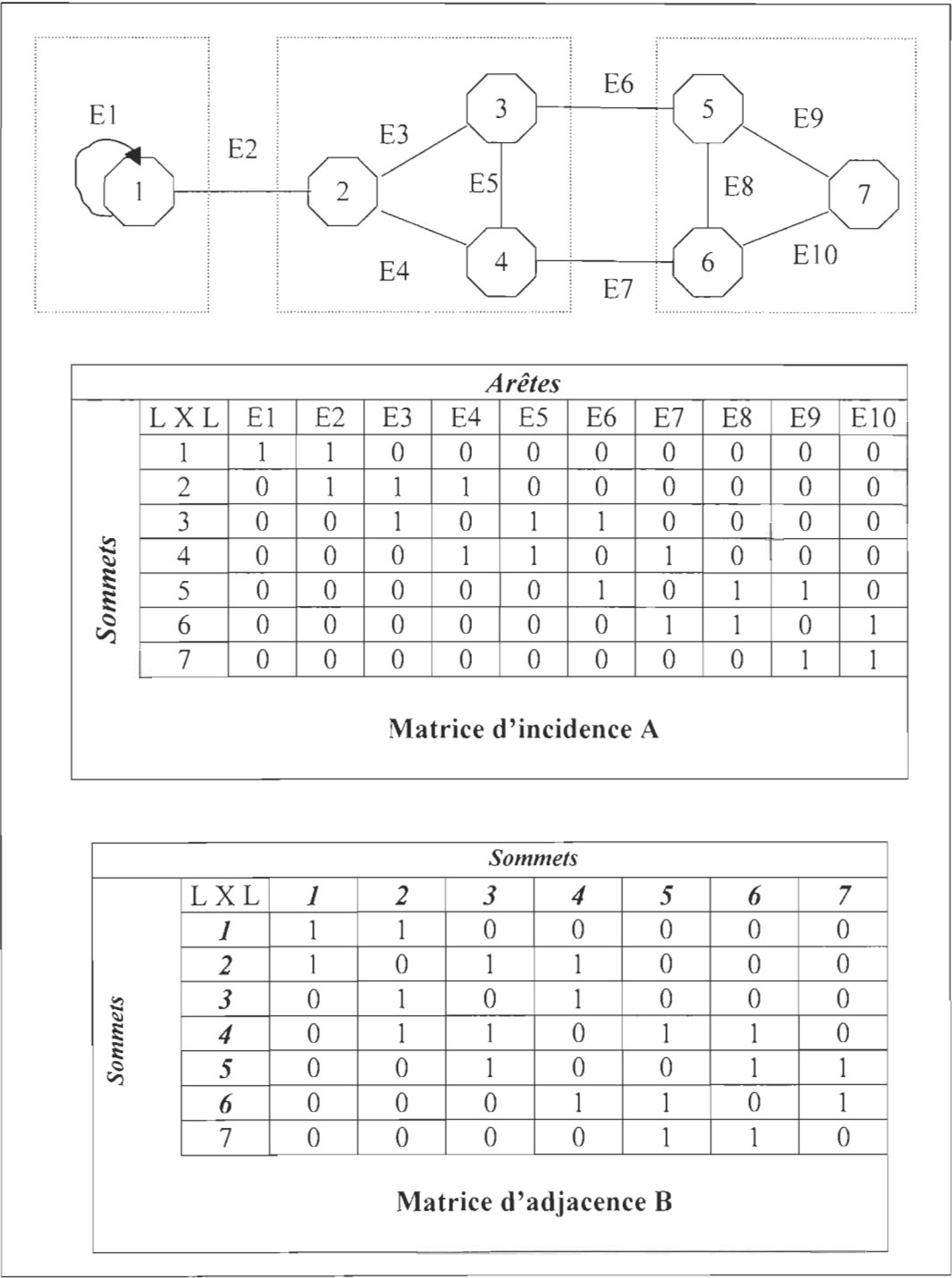


Figure 3.18 : Représentation par matrice d'incidence et d'adjacence

**La deuxième représentation :** Cette représentation est connue par *structure d'adjacence*, tous les sommets reliés à un sommet donné sont énumérés dans une liste d'adjacence de ce sommet. On réalise ceci facilement par des listes chaînées. (figure 3.19)

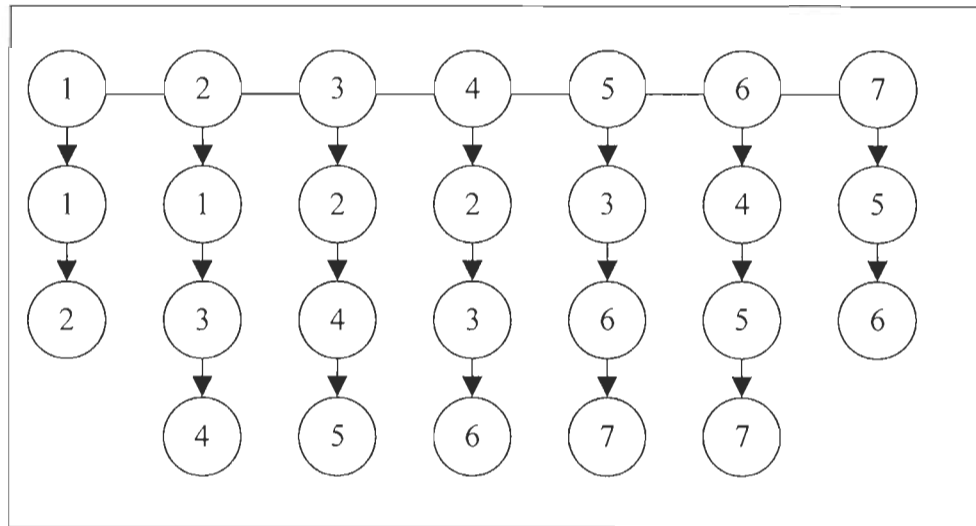


Figure 3.19 : Représentation par structure d'adjacence

Après les différentes définitions et représentations nous pouvons présenter le graphe des éléments fonctionnels de notre exemple par la matrice d'incidence (figure 3.20) et la représentation par la liste d'adjacence (figure 3.21).

		<i>Arêtes</i>					
<i>Sommets</i>		R1	R2	R3	R4	R5	R6
	[3,1]	0	0	0	C	C	J
	[2,2]	0	C	0	0	0	0
	[2,3]	C	0	0	0	0	0
	[2,4]	0	0	0	C	0	0
	[2,10]	0	0	C	0	0	0
	[1,2]	0	0	0	0	0	0
	[1,5]	C	0	0	0	C	0
	[1,6]	0	0	C	0	0	0
	[1,7]	0	C	0	0	0	0

Note : C = Contact & J = Condition fonctionnelle

Figure 3.20 : Représentation par matrice d'incidence

Dans une structure d'adjacence seulement les éléments en contacts et la condition fonctionnelle sont pris en considération. Une remarque importante : la condition fonctionnelle est représentée dans un sens.

La structure suivante est la structure d'adjacence de l'exemple 3.1.

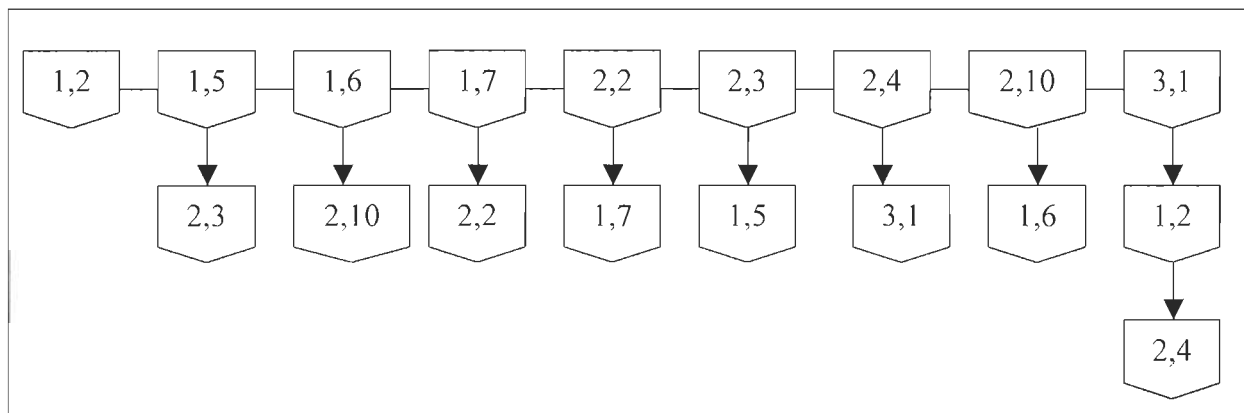


Figure 3.21 : GEF de l'exemple par structure d'adjacence

On peut remarquer que dans cette structure l'ensemble des sommets est la somme de tous les éléments fonctionnels du premier niveau de la structure d'adjacence. Les éléments verticaux forment les contacts et la condition fonctionnelle à respecter.

### 3.2.2.2 Exploitation de la théorie de graphe

En utilisant les définitions citées plus haut (matrice d'incidence, structure d'adjacence etc. ) nous permet d'implanter le graphe d'éléments fonctionnels GEF en utilisant les étapes suivantes :

1. Extraire tous les éléments fonctionnels à partir d'un fichier BREP pour les mémoriser dans une liste simplement chaînée.
2. Éliminer de la liste les éléments qui n'ont pas un contact avec d'autre.
3. Implanter la structure d'adjacence en insérant seulement les éléments qui ont des contacts afin de structurer les contacts. La structure d'adjacence permet d'obtenir facilement les éléments des contacts. Exemple l'élément [1,7] est en contact avec [2,2]. L'implantation d'une telle structure est basée sur la matrice d'adjacence.
4. Implanter la matrice d'incidence en utilisant la structure d'adjacence pour trouver les contacts. L'implantation de la matrice d'incidence permet facilement de trouver les paires entre le contact et les autres éléments dans une même pièce.
5. Une fois la matrice d'incidence obtenue nous pouvons obtenir les chaînes autour d'une condition fonctionnelle.

Pour mieux illustrer les principes précédant, consulter le chapitre 5 exemple d'application sur un mécanisme d'une pompe à engrenage.

### 3.2.2.3 Algorithme d'implantation du GEF

Nous bénéficions de toutes les représentations qui ont été vues dans ce chapitre. Pour garantir l'évolution d'un graphe, il est souvent commode de représenter les ensembles de successeurs et de prédécesseurs par des listes chaînées que l'on appelle GRAPHECONTACT. Les sommets et les têtes de liste sont aussi dans une liste de sommets, représentée de manière contiguë ou chaînée. Cette liste contient tous les sommets du graphe, seulement les éléments des contacts sont en mémoire dans le graphe d'éléments fonctionnels, ce qui permet de gérer conjointement  $L_G$ ,  $R_G$  et  $\psi_G$ . La figure 3.22 montre les listes de successeurs pour le graphe de notre exemple.

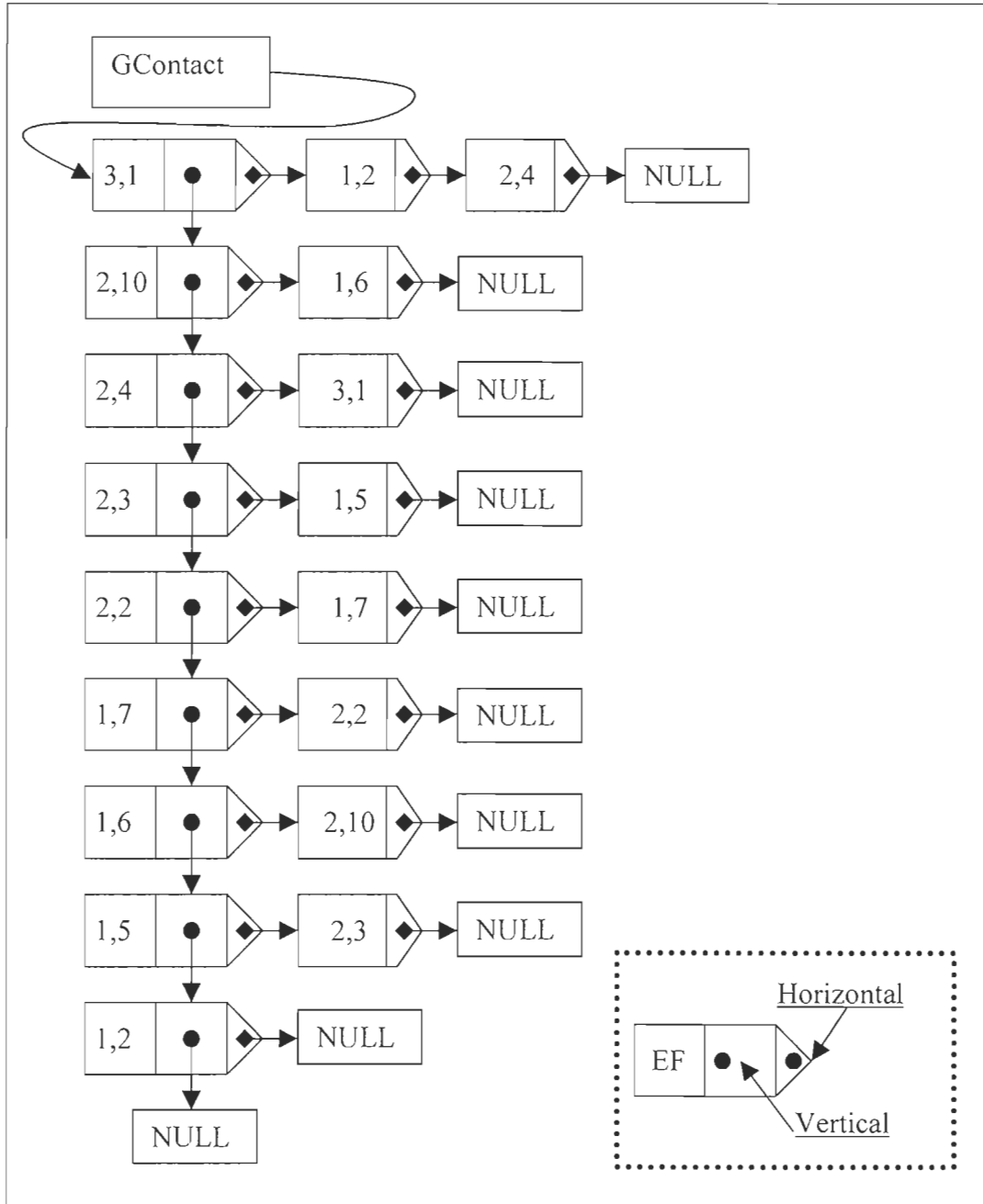


Figure 3.22 : Implantation d'un graphe d'éléments fonctionnels

Nous pouvons remarquer que l'implantation contient deux variables : une verticale qui représente tous les éléments fonctionnels dans le mécanisme en considération et l'autre horizontale qui représente la liste des successeurs de l'élément.

Pour l'implantation du graphe d'éléments fonctionnels il faut tout d'abord définir la structure des données et remplir le tableau 3.4. Ce tableau contient toutes les informations de l'implantation.

Tableau 3.4 : Structure des données de sélection des contacts et condition fonctionnelle

Tableau des sélections				
N# Sélection	Solide #1	Élément#1	Solide #2	Élément#2
1 → N	.....	.....	.....	.....
Tableau de la condition fonctionnelle				
	Solide départ	Élément départ	Solide d'arrivée	Élément d'Arrivée

Nous pouvons maintenant parler de l'implantation de graphe

Tableau 3.5 : Algorithme de l'implantation de graphe d'éléments fonctionnels

```

Void GraphContacts::FaitContact(CptrList &ListeSelect)
{
    Saisir le tableau de sélection d'éléments en contacts
    Saisir le tableau de la condition fonctionnelle

    Ajouter l'élément (Solide départ, Élément départ) dans la liste verticale
    Ajouter l'élément (Solide départ, Élément départ)
        à la liste Horizontale de l'élément (Élément d'arrivée, Élément d'Arrivée)

    Pour chaque numéro de sélection

        Ajouter l'élément (Solide#1,Element#1) dans la liste verticale
        Ajouter l'élément (Solide#2,Element#2) dans la liste verticale

        Ajouter l'élément (Solide#1,Element#1)
            à la liste Horizontale de l'élément (Solide#2,Element#2)
        Ajouter l'élément (Solide#2,Element#2)
            à la liste Horizontale de l'élément (Solide#1,Element#1)

    Fin pour
}

```

**NOTE** : GraphContacts est une classe développée en C++ dont le rôle est de faire les connexions entre les éléments fonctionnels à partir de la liste obtenue auparavant à partir du fichier BREP.

### 3.2.3 Identification des paires d'éléments fonctionnels

Comme était mentionner plus haut que deux éléments fonctionnels peuvent former une paire d'éléments fonctionnels si les deux éléments appartiennent à une même pièce et ont au moins un contact avec un autre élément dans une autre pièce.

En utilisant les contacts dans le GEF nous pouvons déterminer dans chacune de pièce l'ensemble des paires d'éléments fonctionnels. Pour plus de détail voir le chapitre 5 (exemple d'application).

**DetListeSortie**( Data, ListeS) est une fonction à deux paramètres. Le premier est l'élément « Data » d'une PEF et le deuxième paramètre est la liste des éléments qui peuvent former une PEF avec le « Data ».

Tableau 3.6 : Algorithme d'identification des paires d'éléments fonctionnels

```
Void GraphContacts:: DetListeSortie(Mere *Data,CPtrList &ListePaire)
{
    Déterminer la pièce à laquelle l'élément appartiennent
    Tant que le vertical du graphe est différent de NULL
        Si l'élément vertical et Data appartiennent à la même pièce
            Alors
                Ajouter vertical à la liste des paires(ListePaire)
        Fin tant que
    }
```

Selon la théorie de graphe le nombre maximum de connexion doit satisfaisant la relation suivante :

$$NombreConnexion = \frac{N \bullet (N - 1)}{2}$$

Où N est le nombre d'éléments dans une pièce

Ce nombre de connexions est le nombre de paires d'éléments dans une pièce donnée. Pour notre exemple le nombre de connexions possibles dans la pièce 2 est égal à 6.

Dans une pièce à N éléments, l'élément peut former une paire avec les autres éléments de maximum (N-1). Par exemple une pièce de 4 éléments fonctionnels, chaque élément à un contact avec un autre d'une autre pièce. Selon notre définition pour une paire d'élément il faut que les deux éléments aient au moins un contact avec d'autres, alors chaque élément peut former 3 paires avec les autres éléments de la même pièce.

Une fois que les paires d'éléments fonctionnels sont identifiées, il sera possible d'associer à chacune des paires, des tolérances en consultant le tableau 3.6

### 3.2.4 Affectation de tolérances sur une paire d'éléments fonctionnels

Un simple algorithme est appliqué sur toutes les paires d'éléments fonctionnels dans chacune des pièces dans le graphe pour affecter les différents types de tolérances en consultant les possibilités déterminées auparavant (tableau 3.3).

La liste d'éléments nous fournit les différents types de surface (Plan, courbe, etc.) et pour chaque type nous avons l'accès au centre et le vecteur normal à ce type de surface. Un simple calcul peut fournir le type de tolérance à imposer. Cette tâche est automatiquement réalisée par le système.

Le graphe d'éléments fonctionnels de l'exemple devient, après l'affectation des tolérances, comme à la figure (3.23).

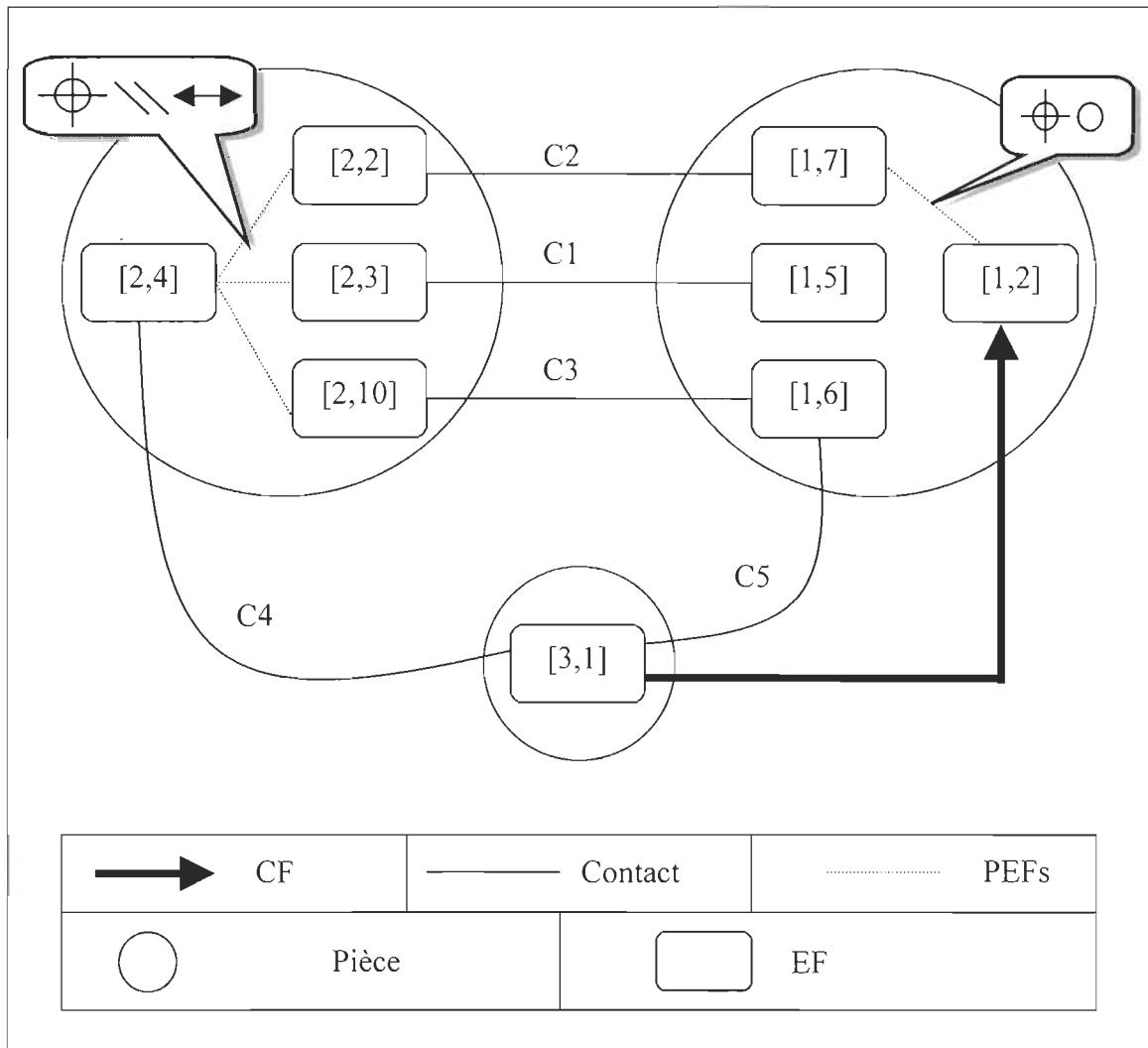


Figure 3.23 : Graphe d'éléments fonctionnels après l'affectation des tolérances



### 3.2.5 Existence et dénombrement des chemins dans un graphe

**Définition d'un chemin :** Selon A. Kaufman [1], Un chemin est une séquence ( R1, R2, R3...) d'arêtes telles que l'extrémité terminale de chaque arête de la séquence correspond à l'extrémité initiale de la suivante. Un chemin peut être simple, composé ou élémentaire. La longueur d'un chemin est égale au nombre d'arêtes dans un chemin.

Dans une chaîne de cotes, nous nous intéressons plus au chemin simple qui ne contient pas deux fois la même arête. Les arrêtes dans notre graphe d'éléments fonctionnels sont des contacts et des paires d'éléments fonctionnels.

Une chaîne de cotes débute par le départ de la condition fonctionnelle et se termine par l'élément d'arrivée de la condition fonctionnelle. Il est intéressant de connaître le nombre maximum de chaînes de cotes dans un graphe d'éléments afin de valider notre algorithme.

Soit le graphe de l'exemple précédent (figure 3.18 )  $G = \{ L_G, R_G, \psi_G \}$ . Par définition, la matrice d'incidence nous donne l'existence des chemins de longueur 1. En se basant sur la théorie combinatoire appliquée sur le graphe [1], nous pouvons calculer successivement :

$$\begin{aligned} [A]^2 &= [A] \bullet [A], \\ [A]^3 &= [A]^2 \bullet [A] \\ &\dots\dots\dots \\ [A]^r &= [A]^{r-1} \bullet [A], \text{ Où } \bullet \text{ représente la multiplication matricielle ordinaire} \end{aligned}$$

$[A]$  : Donne les nombres de chemins de longueur ( 1 ) de chaque  $X_i$  et  $X_j$

$[A]^2$  : Donne les nombres de chemins de longueur ( 2 ) de chaque  $X_i$  et  $X_j$

$\dots\dots\dots$

$[A]^r$  : Donne les nombres de chemins de longueur ( r ) de chaque  $X_i$  et  $X_j$

Pour vérifier nos chaînes, nous avons conçu un programme qui reçoit une matrice d'incidence pour les chemins de longueur (1) et la longueur désirée ( r ). Le programme nous fournit la matrice pour le nombre de chemins de longueur ( r ) donné à l'entrée. La figure 3.24 montre un exemple des résultats pour le graphe de la figure 3.18.

Pour connaître le nombre de chaîne autour d'une condition il faut tout simplement sélectionner la longueur voulue, le programme donne la matrice totale. C'est un outil de vérification non de développement. Ce programme ne génère pas les chaînes mais par contre donne une idée du résultat à obtenir.

#### Notes :

- ♦ Le case dans la matrice d'incidence représente les contacts entre les différents éléments fonctionnels ou l'existence d'une paire d'éléments fonctionnels. Les colonnes et les lignes représentent les éléments fonctionnels dans le mécanisme.
- ♦ Nous pouvons exécuter le programme plusieurs fois pour vérifier si notre graphe peut obtenir des chaînes à une longueur précise par exemple plus petit que le nombre de pièces dans le mécanisme.

La figure 3.24 montre un écran d'exécution de l'exemple de la figure 3.18.

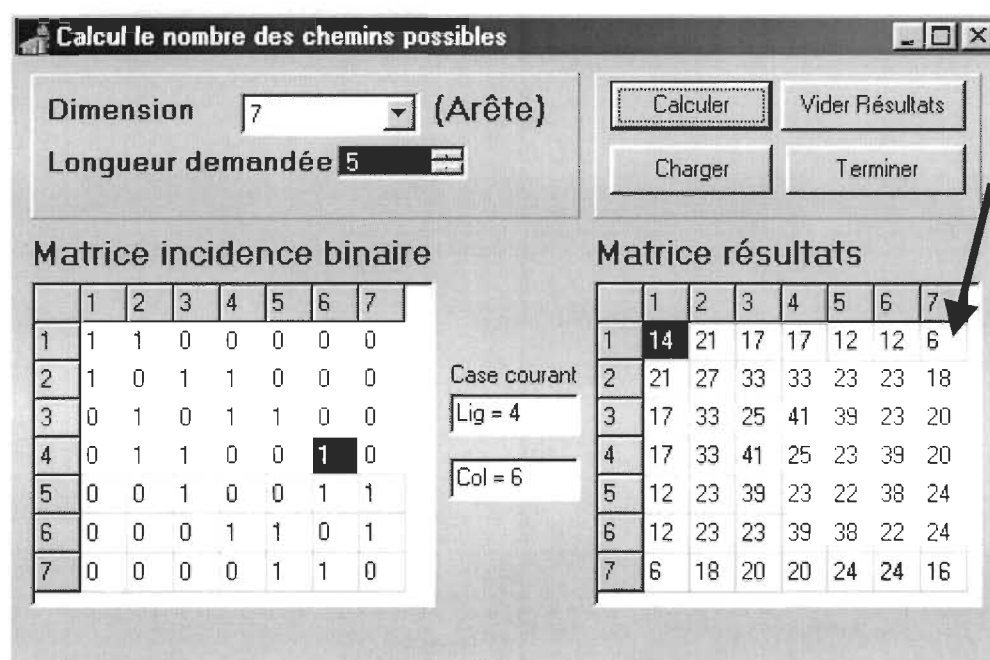


Figure 3.24 : Le nombre de chemins possibles de la figure 3.18

Nous pouvons remarquer qu'il existe 6 chemins possibles entre les sommets 1 et 7.

### 3.3 GÉNÉRATION DE L'ARBRE DES ÉLÉMENTS FONCTIONNELS (AEFS)

#### 3.3.1 Définition

A. Kaufman [1] définit l'arbre comme suit :

Un graphe  $G = \{L_G, \Gamma_G\}$  est une arborescence de racine  $X_0$ , si et seulement si :

- $\exists ! X_0 \in L_G : \Gamma^{-1}\{X_0\} = \emptyset$ .
- $\forall X_i \in L_G, X_i \neq X_0 : |\Gamma^{-1}\{X_i\}| = 1$ .
- Le graphe ne contient aucun circuit.

Autrement dit : il existe un et un seul sommet appelé racine où n'arrive aucun arc; il arrive un et un seul arc aux autres sommets, enfin le graphe est sans circuit.

À première vue on peut dire que cette définition exclut le graphe d'éléments fonctionnels obtenu auparavant. Mais si nous examinons comme il le faut l'organisation de ces éléments dans le GEF, nous pouvons remarquer que seulement les éléments qui ont des contacts sont connectés avant le parcours de ce graphe. Ensuite, une fois la condition fonctionnelle imposée et le parcours du graphe débuter les paires d'éléments sont en connexion ce qui minimise le nombre de connexions dans chaque pièce parce qu'on ne peut avoir deux paires successives dans une chaîne d'une même pièce. Dans ce cas on ne peut avoir aucun circuit dans ce graphe. (Voir le chapitre 5 exemple d'application).

Nous pouvons conclure que le graphe obtenu est une arborescence à une racine de l'élément de départ de la condition fonctionnelle et chaque feuille de cet arbre se termine par l'élément d'arrivée de cette CF. Pour obtenir l'arbre des éléments fonctionnels il suffit tout simplement de parcourir le GEF obtenu à partir du premier élément jusqu'au deuxième de la condition fonctionnelle.

### 3.3.2 Représentation graphique

La structure d'arbre est la structure de données primordiale en informatique. Diverses variantes sont utilisées : arbres binaires, arbres binaires complets, arbres généralisés, forêts d'arbres.

Parmi les différentes structures d'arbres possibles, l'arbre binaire est la structure la plus intéressante que nous pouvons utiliser. Il existe de nombreuses manières pour représenter l'arbre binaire : arbre ordonné, arbre de tri, arbre syntaxique, arbre généalogique, arbre AVL, arbre B et autres.

Un arbre binaire est composé de **nœuds**. Chaque nœud pointe en général sur deux autres nœuds : ses **fils**. En fait un nœud peut n'avoir qu'un seul fils ou même aucun : dans ce cas c'est une **feuille**. Le nœud qui n'a pas de père est la **racine**. Il suffit d'avoir d'un pointeur vers la **racine** pour pouvoir parcourir l'arbre en entier. Un nœud et ses **descendants** forment un **sous arbre** ou **branche**. (Figure 3.25)

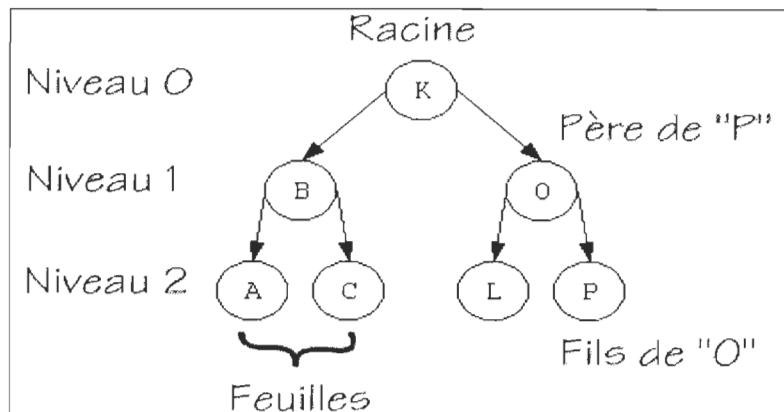


Figure 3.25 Représentation de l'arbre binaire

Les arbres binaires sont bien adaptés pour les problèmes de tris, les arbres AVL sont utilisés pour compléter les arbres binaires et les arbres B et *n-Aire* sont utilisées dans les domaines des bases de données, etc.

**Insertion dans un arbre binaire :** Voyons maintenant comment rajouter un nœud à l'arbre. Il faut d'abord rechercher sa place : pour que notre arbre soit un arbre binaire il faut que tous les fils droits soient inférieurs à leurs pères et que tout les fils gauches leurs soient supérieurs. Pour satisfaire cette règle, on utilise un pointeur qui parcourt l'arbre en comparant la valeur du nouvel élément à celui des nœuds rencontrés. On sait que l'on a trouvé une place libre lorsque l'on rencontre un pointeur sur « **nil** ». On le remplace alors par un pointeur sur le nouveau nœud.

Nous pouvons voir comment organiser (figure 3.26) les données lorsque les éléments de la cette liste {25, 30, 12, 26, 10, 40, 38} sont enregistrés dans l'ordre.

La figure suivante montre le principe d'insertion dans un arbre binaire.

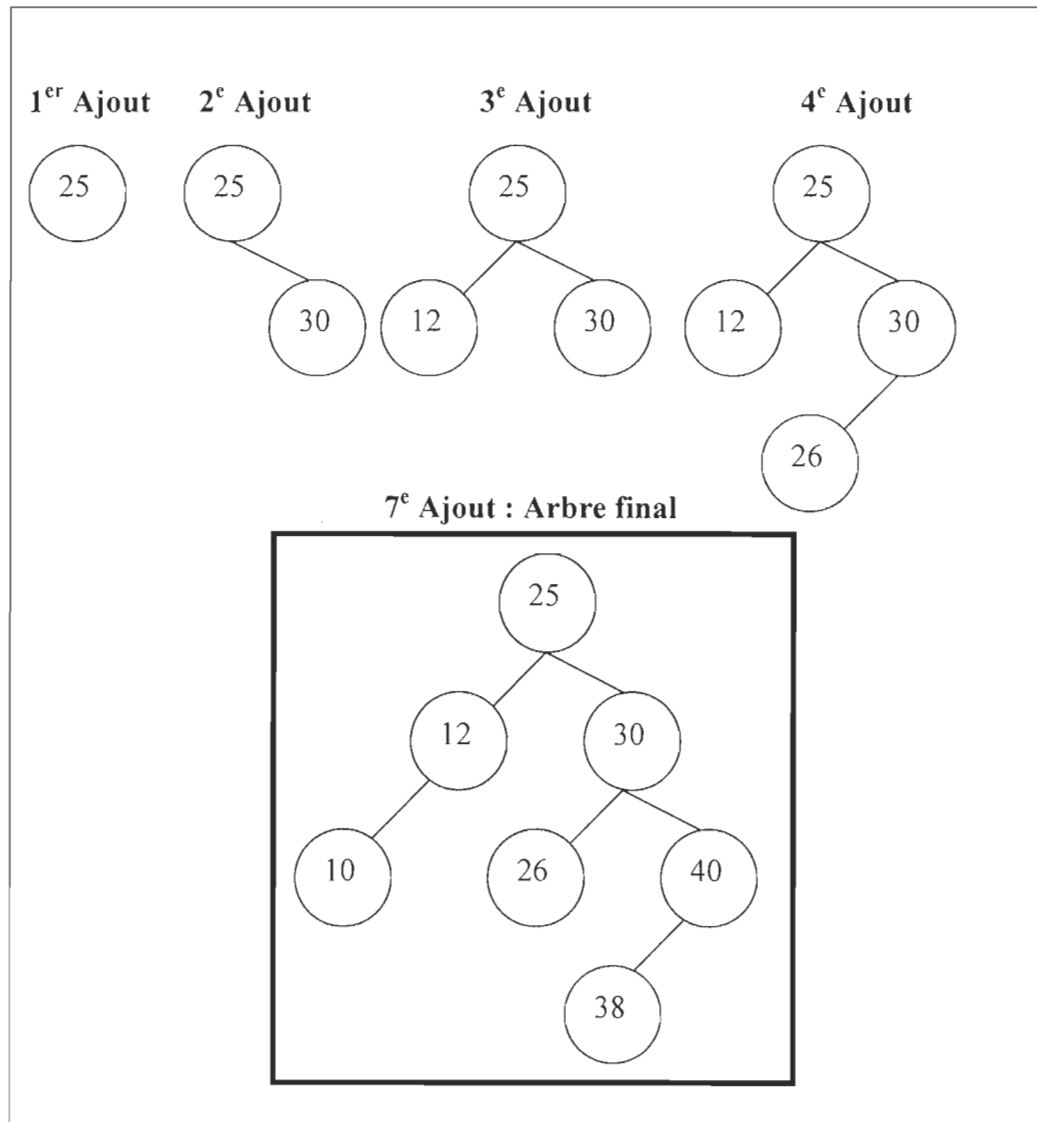


Figure 3.26 Insertion dans une structure d'arbre binaire

Il n'y a pas une représentation bien adaptée à notre problème, mais nous pouvons modifier la structure de l'arbre binaire.

Si nous examinons le graphe d'éléments fonctionnels obtenu auparavant (exemple figure 3.17) en éliminant la condition fonctionnelle nous pouvons remarquer que ce graphe est un arbre binaire mais à plusieurs fils. Autrement dit que chaque élément dans la liste peut avoir plusieurs liens avec d'autre (contacts ou PEFs).

Nous pouvons adapter la structure de l'arbre binaire pour obtenir une structure d'arbre binaire modifié. La section suivante montre les changements apportés sur la structure de l'arbre binaire.

### 3.3.3 Implantation

Nous avons modifié la structure de l'arbre binaire en utilisant deux variables : une verticale pour tous les éléments fonctionnels en contacts, et l'autre pour les possibilités de former des paires d'éléments. La figure 3.27 montre la représentation de notre structure binaire modifiée.

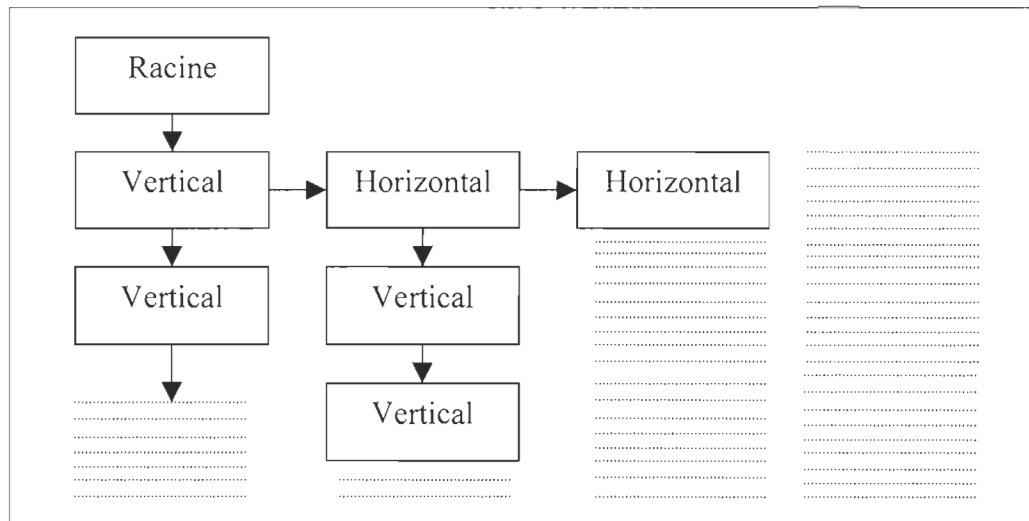
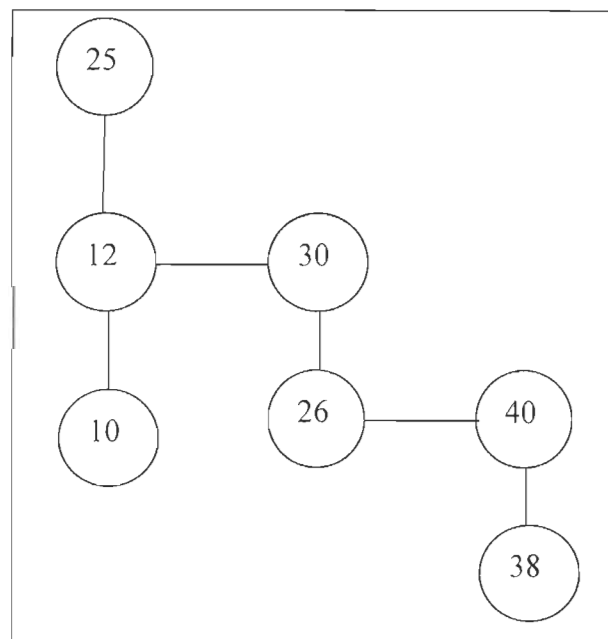


Figure 3.27: Représentation graphique de l'arbre d'éléments fonctionnels

L'arbre est constitué d'une seule racine et de plusieurs nœuds (figures 3.27 & 3.28). La racine est l'élément de départ de la condition fonctionnelle. Les nœuds sont des paires d'éléments fonctionnels, autrement dit les deux appartiennent à une même pièce et chaque élément est en contact avec un autre élément d'une autre pièce. Ceci donne les chaînes des cotes.

La représentation de la figure 3.26 devient par la nouvelle structure comme suit (figure 3.28)



3.28 Exemple d'arbre binaire modifié

Nous avons utilisé deux méthodes pour parcourir le graphe. La première consiste à explorer le graphe dans un but de visualisation à l'écran, et l'autre consiste à mémoriser tous les chemins possibles dans un tableau pour consultation à l'écran ou pour l'imprimer ou sauvegarder les chaînes de cotes dans un fichier pour les consulter plus tard.

L'algorithme suivant (Tableau 3.7) montre comment on peut parcourir le GEF en utilisant la contrainte que le chemin ne devrait passer plus d'une fois par une arête et un sommet. Le chemin est une chaîne qui est expliquée en théorie de graphe par une chaîne *hamiltonienne*.  
[1][16][17][20]

Tableau 3.7 : Algorithme d'arbre d'éléments fonctionnels.

```
void ArbreChaine::FaitTableaux(Tselections Selections)
{
    Déclarer un tableau de paires d'éléments fonctionnels à deux dimensions
    Déclarer une liste de positions avec un enregistrement à deux variables (Ligne,
    colonne)

    Déterminer à partir du graphe d'éléments fonctionnels les éléments des contacts
    dans la même pièce de l'élément de départ de la condition fonctionnelle. En
    consultant la structure d'adjacence

    Pour chaque contact avec l'élément de départ
        Combiner l'élément de départ de la CF à chaque élément de contact
        Ajouter la paire combinée dans le tableau des paires dans une nouvelle ligne
        Ajouter la position de la ligne et de la colonne à la fin de la liste de positions
    Fin pour
    Tant que la liste de positions n'est pas vide
        Prendre le premier enregistrement de la liste de positions
        Déterminer la liste des contacts du deuxième élément de l'enregistrement

        Pour chaque élément de contact obtenu
            Combiner le deuxième élément de l'enregistrement à chaque élément
            de contact
            Ajouter la paire combinée dans le tableau des paires dans une
            nouvelle ligne
            Ajouter la position de la ligne et de la colonne à la fin de la liste de
            positions
        Fin pour

        Effacer le premier enregistrement de la liste de position
    Fin tant que
}
```

**NOTE** : ArbreChaine est une classe, développée en C++, son rôle est le parcours du graphe d'éléments fonctionnels pour obtenir l'AEFs. ArbreChaine permet la mémorisation de chaque branche de l'AEFs dans un tableau pour l'affichage.

L'affichage de l'arbre d'éléments fonctionnels a été réalisé en deux étapes. La première étape consiste à mémoriser le graphe d'éléments fonctionnels, dans une liste arborescente qui contient : verticalement les contacts entre les pièces (flèches vers le bas) et horizontalement les paires d'élément dans une même pièce. La deuxième étape consiste à utiliser le contrôle Active X «CTRTREE » en VC++ pour la visualisation de l'arborescence.

Le parcours du graphe d'éléments fonctionnels de la figure 3.16 nous donne l'arborescence suivante qui devient l'arbre des paires d'éléments fonctionnels (figure 3.29).

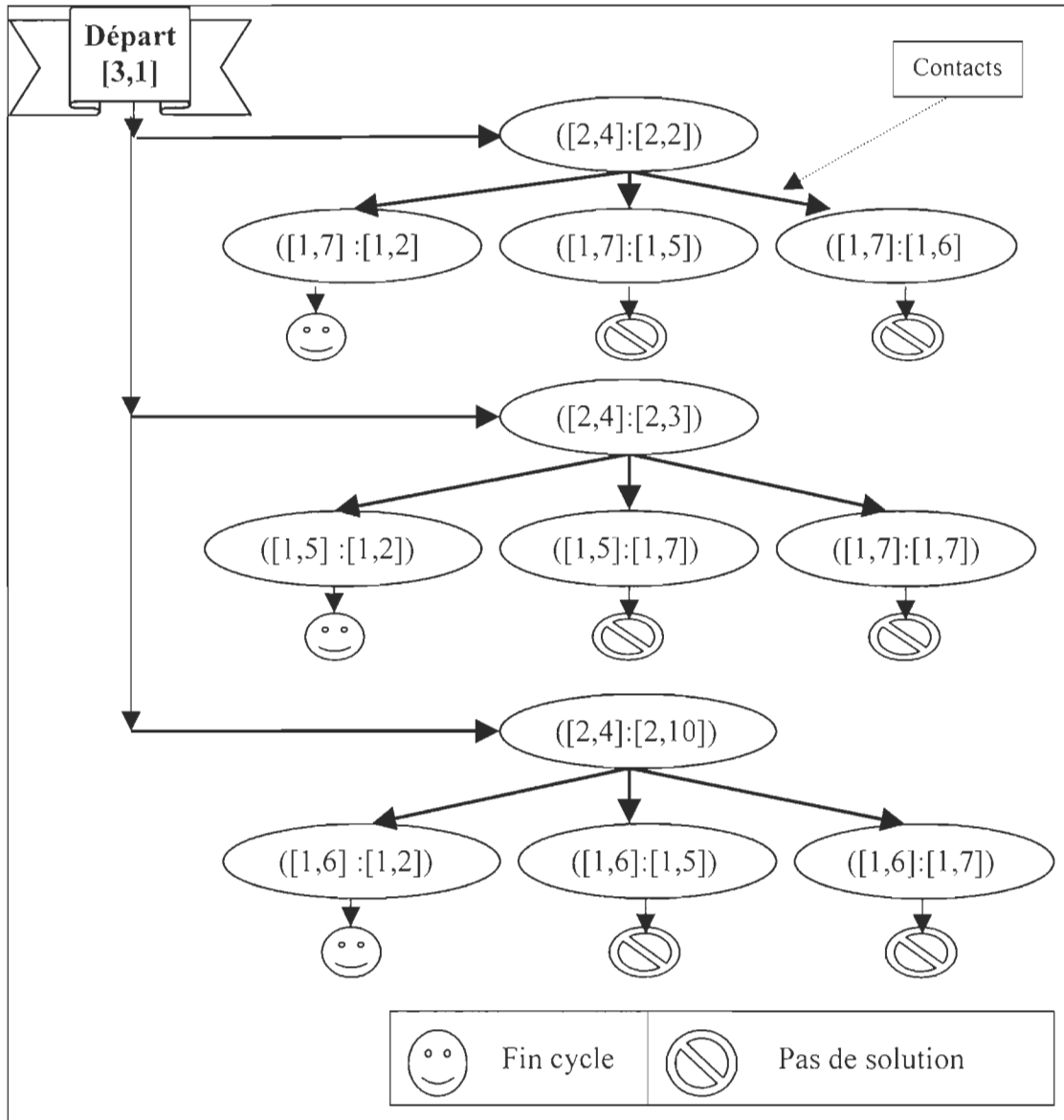


Figure 3.29 : Arbre des paires d'éléments fonctionnels de l'exemple

Notes :

- ♦ l'élément [2,2] forme contact avec [1,7] (verticalement)
- ♦ l'élément [1,7] forme une paire avec [1,2], [1,5] et [1,6] (horizontalement)



## CHAPITRE 4

### ENVIRONNEMENT DU LOGICIEL

---

#### 4.1. LA STRUCTURE DU LOGICIEL

#### 4.2. PRÉSENTATION GÉNÉRALE DU LOGICIEL.

##### 4.2.1. Menu Fichier

4.2.1.1. L'option : Nouveau

4.2.1.2. L'option : Visualiser

##### 4.2.2. Menu : Graphe d'éléments fonctionnels

4.2.2.1. L'option : Sélectionner des contacts

4.2.2.2. L'option : Sélectionner d'une condition fonctionnelle

4.2.2.3. L'option : Générer le graphe d'éléments fonctionnels

##### 4.2.3. Menu : Arbre des paires d'éléments

4.2.3.1. L'option : Construction Arbre

4.2.3.2. L'option : Résultats des chaînes de cotes

4.2.3.3. L'option : Détail des chaînes de cotes



#### 4.1. LA STRUCTURE DU LOGICIEL

Analyse en Tolérancement Assisté par Ordinateur (TAO) est un programme qui permet de déterminer les chaînes de cotes tridimensionnelles d'un mécanisme contenant au moins deux pièces. Ce programme s'exécute sur une plate-forme de Windows 95 ou plus avec au moins 32 M de mémoire, il contient des menus déroulant et des sous-menus faciles à utiliser.

La structure interne du programme (Analyse en TAO) comporte trois niveaux. Le premier niveau s'occupe de la saisie des données, le deuxième du traitement et le dernier de l'affichage du résultat. La figure 4.1 montre les trois niveaux du programme.

La saisie des données passe par une première étape qui est la lecture du fichier en trois dimensions modélisé par le fichier BREP, ensuite par la sélection des contacts et de la condition fonctionnelle entre les éléments fonctionnels des différentes pièces.

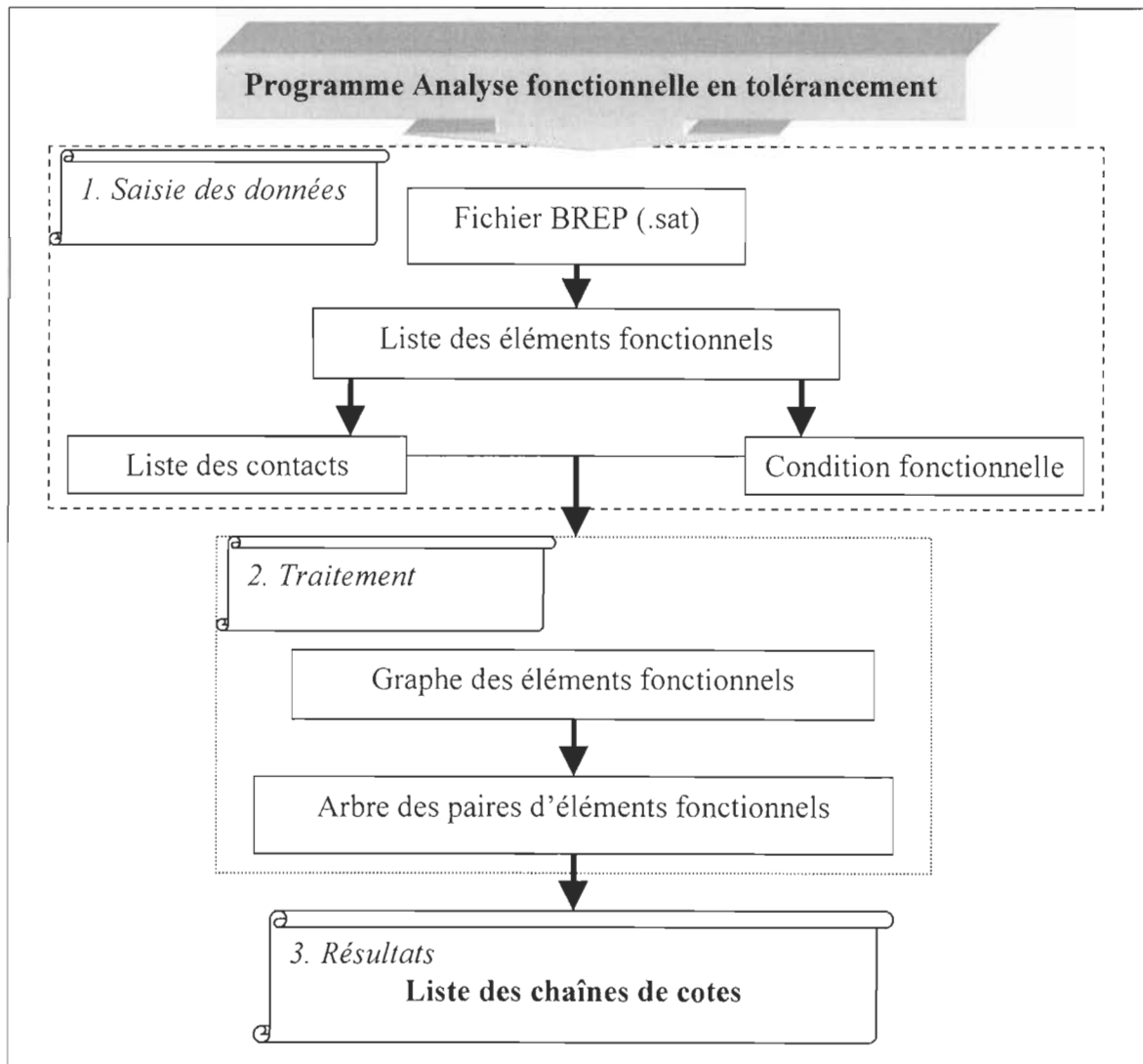


Figure 4.1 : Structure interne du logiciel (Analyse TAO).

## 4.2. PRÉSENTATION GÉNÉRALE DU LOGICIEL.

Pour exécuter le programme il suffit de cliquer deux fois sur l'icône (Analyse TAO). Immédiatement après le lancement du programme, un menu principal (figure 4.2) offre à l'utilisateur quatre menus qui peuvent être sélectionnés à l'aide de la souris ou des caractères soulignés. Ces différents menus seront illustrés ultérieurement.

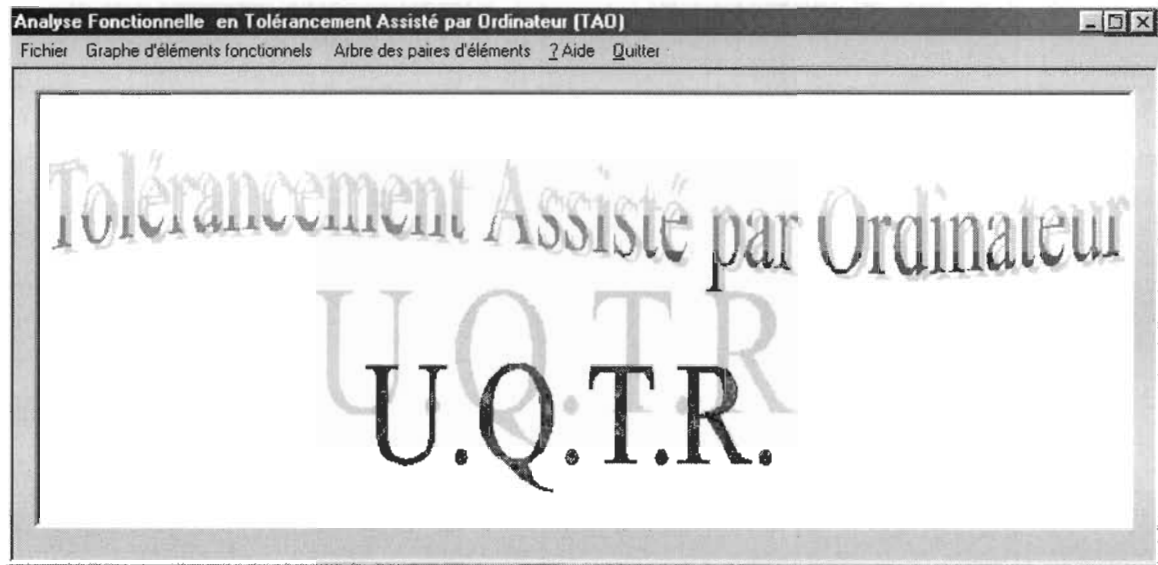


Figure 4.2 : Écran d'accueil de (TAO)

On peut obtenir dix différents sous menus à partir du menu principal (Figure 4.3) :

<b>◆ Fichier</b> Fichier Nouveau Visualiser Quitter	<b>◆ Graphe d'éléments fonctionnels</b> Graphe d'éléments fonctionnels Sélections des contacts Sélection d'une condition fonctionnelle Générer le graphe d'éléments
<b>◆ Aide</b> ? Aide Rubrique À Propos	<b>◆ Arbres des paires d'éléments</b> Arbre des paires d'éléments Construction Arbre Résultats des chaînes de cotes Détails des chaînes de cotes

Figure 4.3 : Les différents sous menus dans (TAO)

Nous allons aborder dans les sections suivantes les fonctionnalités des différents sous-menus du programme.

#### 4.2.1. Menu : Fichier

Cette option contient trois sous menus. Les options **N**ouveau, **V**isualiser et **Q**uitter. L'option **Q**uitter sert tout simplement à quitter définitivement le programme, une validation est prévue avant de terminer la session du travail.

##### 4.2.1.1 L'option : Nouveau

Sert à ouvrir un nouveau fichier « .sat » à analyser. Le classique dialogue d'ouverture du fichier apparaît (figure 4.4), et tous les fichiers de type « .sat » sont accessibles ainsi que les différents répertoires.

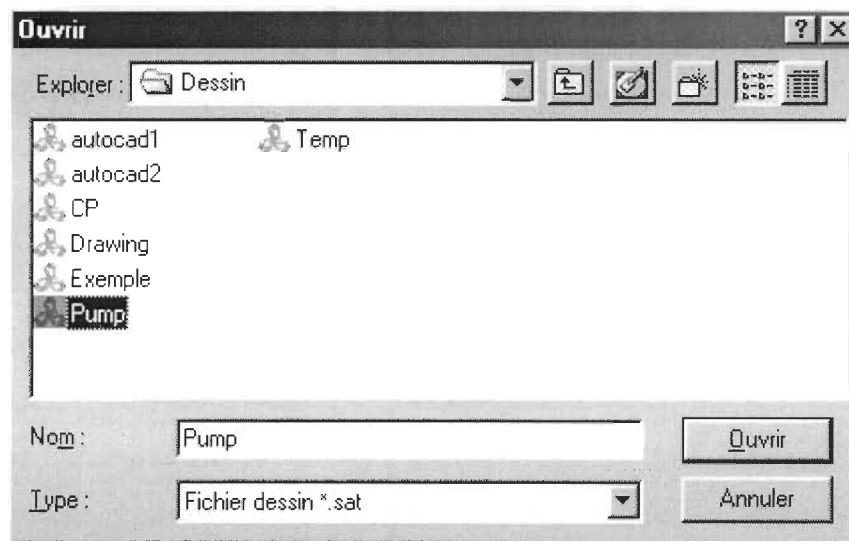


Figure 4.4 : Écran ouvrir un nouveau fichier.

Choisissez le fichier sur lequel vous voulez travailler et ensuite cliquez sur la commande **<Ouvrir>**, le programme charge alors le fichier choisi. Le dessin et ses éléments seront affichés dans la fenêtre suivante (figure 4.5).

La fenêtre 4.5 montre l'écran d'un nouveau fichier à analyser

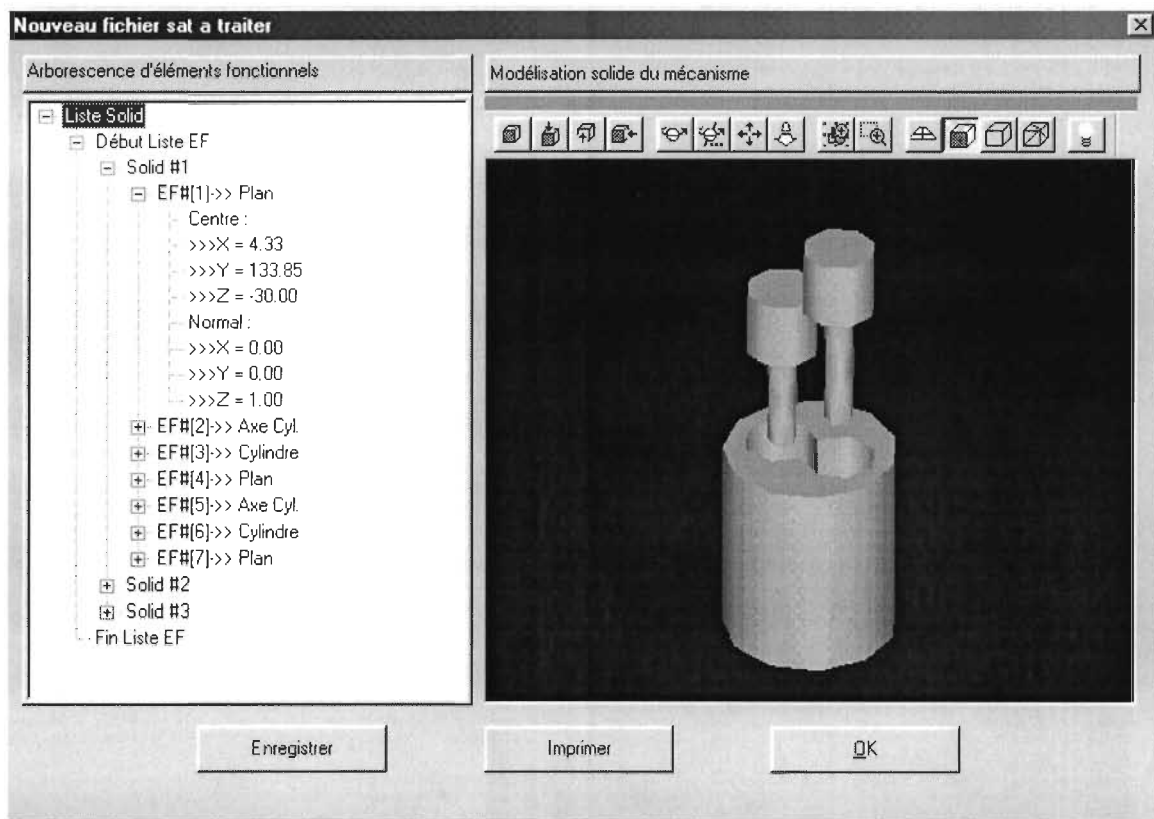


Figure 4.5 : Écran d'un nouveau fichier à analyser.

La fenêtre précédente (figure 4.5) est divisée en deux parties :

- **Arborescence d'éléments fonctionnels** : dans cette section vous avez tous les solides dans le mécanisme avec leurs éléments fonctionnels, leurs orientations et leurs caractéristiques. Par exemple pour un élément plan vous aurez le vecteur normal et le centre de cet élément plan. Pour un cylindre vous aurez en plus de ces informations le rayon et la hauteur.
- **Modélisation solide du mécanisme** : cette partie contient différentes options qui se trouvent en haut du dessin. Ces options sont : Rotation dynamique, Visualiser les différentes vues : gauche, haut, droite et d'autres options.

Parmi les autres options importantes, figurent celles d'impression <Imprimer> et d'enregistrement <Enregistrer> des éléments fonctionnels dans un fichier texte. L'option <OK> ferme la fenêtre pour retourner au menu principal.

#### 4.2.1.2. L'option : Visualiser :

Cette option sert à visualiser le dessin choisi dans l'option nouveau si aucune autre option n'est déjà activée. Cette option ne fournit aucune information à l'exception de la visualisation à l'écran.

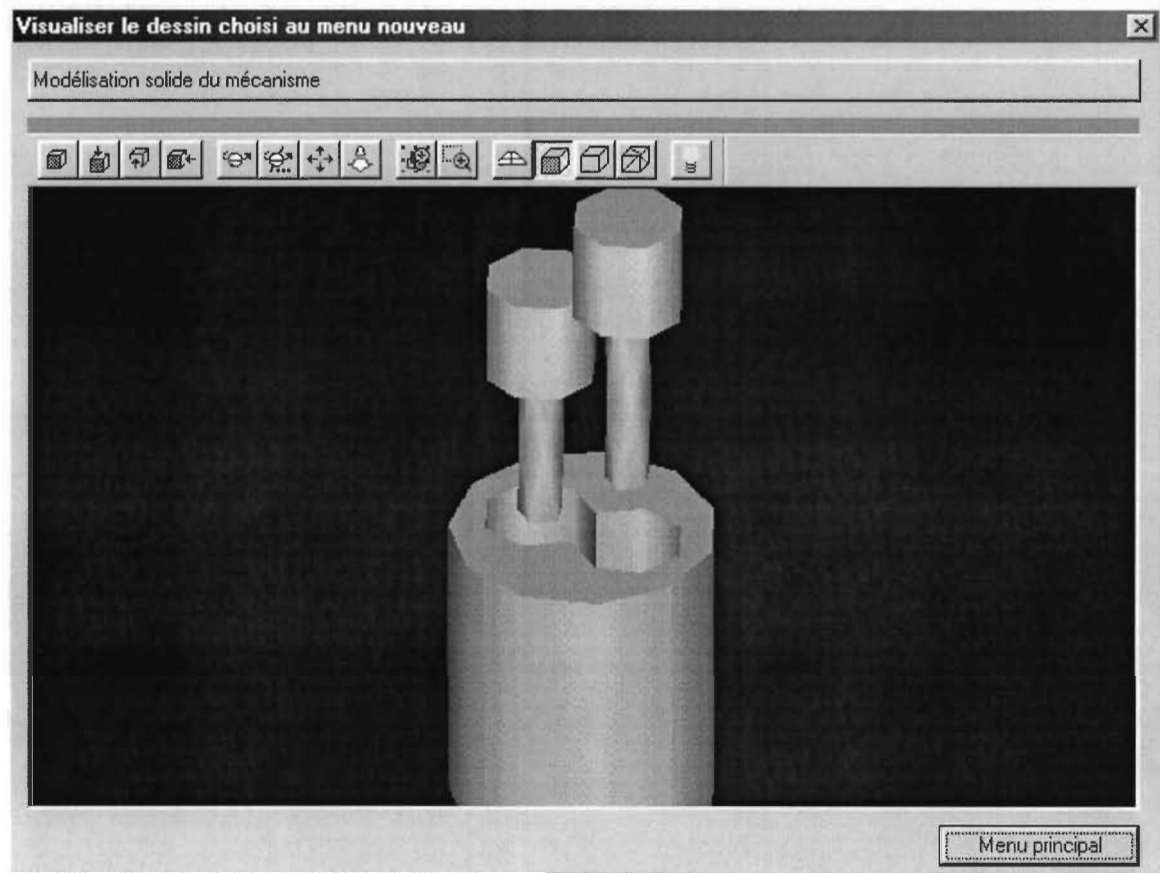


Figure 4.6 : Écran visualisation d'un dessin.

Les différentes options en haut du dessin permettent de :

- ◆ Visualiser les différentes vues du dessin ;
- ◆ Visualiser le dessin en solide, en fil de fer ou en modélisation surfacique ;
- ◆ Agrandir (ZOOM) une partie d'un dessin ;
- ◆ Réduire une partie d'un dessin ;
- ◆ Déplacer le dessin ;
- ◆ Etc.

#### 4.2.2. Menu : Graphe d'éléments fonctionnels

Le menu «Graphe d'éléments fonctionnels» contient trois options pour générer le graphe d'éléments fonctionnels afin d'obtenir l'arbre. Les trois options sont :

- ◆ Sélection des contacts ;
- ◆ Sélection d'une condition fonctionnelle ;
- ◆ Générer le graphe d'éléments fonctionnels.

##### 4.2.2.1. L'option : Sélectionner des contacts

L'option <**Sélectionner des contacts**> permet à l'utilisateur de sélectionner tous les éléments fonctionnels en contact dans différentes pièces.

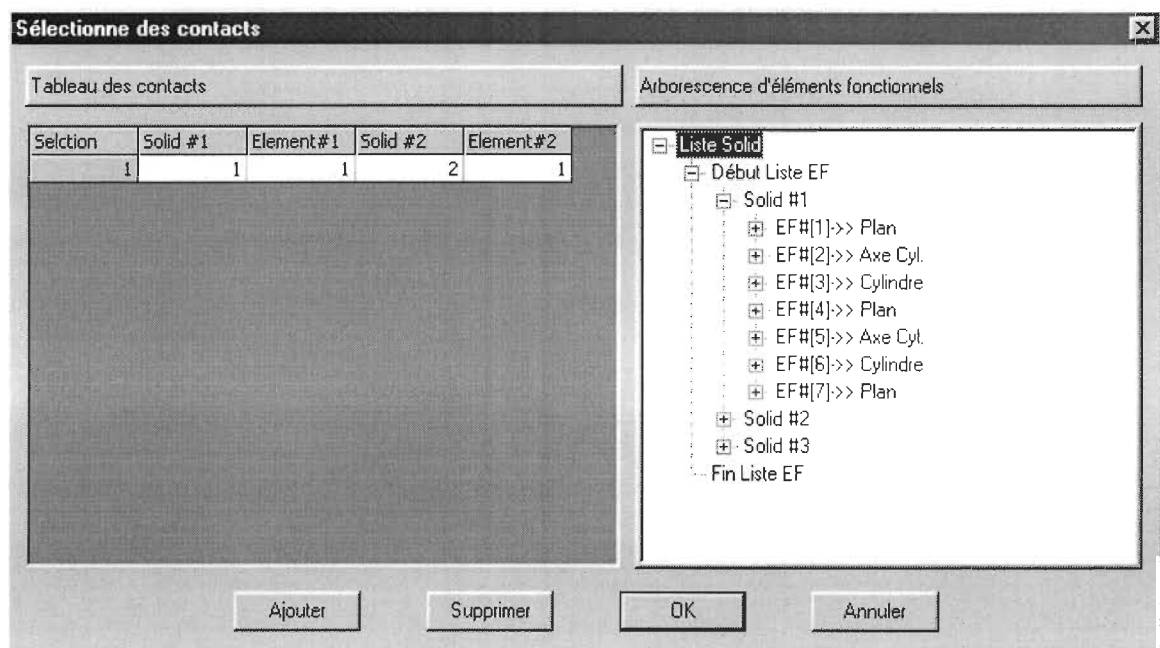


Figure 4.7 : Écran Sélectionner des contacts.

La fenêtre (figure 4.7) est divisée en deux parties.

- ◆ **Partie gauche** : la partie gauche est un tableau qui contient tous les contacts, pour remplir ce tableau il faut tout simplement cliquer sur l'option <**Ajoutez**> pour activer la fenêtre <**Sélectionner un contact**> (figure 4.9), les informations sélectionnées de la fenêtre <**Sélectionner un contact**> vont apparaître dans la fenêtre <**Sélectionner contacts**>. Il faut répéter cette opération pour toutes les sélections possibles.
- ◆ **Partie droite** : la partie droite contient l'arborescence des éléments fonctionnels de chaque solide dans le mécanisme en question.



Deux commandes <Ajouter> et <Supprimer> sont utiles pour sélectionner les contacts.

#### <Supprimer>

**Supprimer** sert à supprimer un élément sélectionné par erreur. Il suffit tout simplement de choisir la ligne à supprimer dans la fenêtre de suppression (figure 4.8).

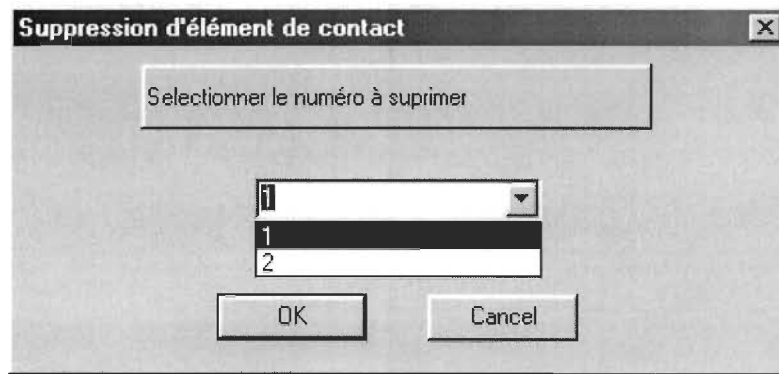


Figure 4.8 : Écran d'une suppression de contact.

#### <Ajouter>

**Ajouter** sert à remplir le tableau des contacts dans la fenêtre de la figure 4.7. Un simple appui sur l'option <Ajouter> dans la fenêtre précédente (figure 4.7) et vous aurez cette fenêtre :

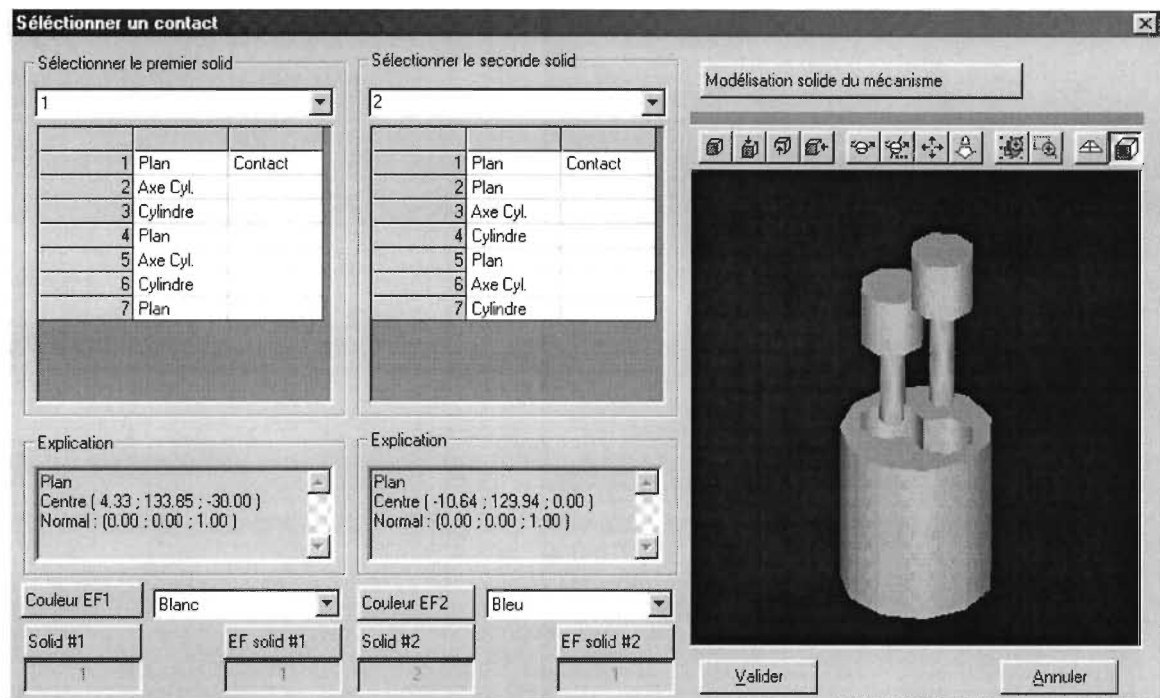


Figure 4.9 : Écran ajouter un contact.

La fenêtre 4.9 permet de choisir deux éléments en contacts entre des pièces différentes. Des informations se trouvent, dans les zones (Explication) de la fenêtre, servant à faciliter les choix des éléments concernés. La fenêtre de la sélection des contacts est divisée en trois zones.

- *Zone haut gauche* : cette zone est réservée pour les choix d'éléments fonctionnels. D'abord il faut choisir la pièce dans la liste déroulante et ensuite l'élément. Des informations apparaissent dans la zone gauche en bas concernant l'élément choisi.
- *Zone bas gauche* : cette zone est réservée pour les informations sur les éléments choisis. Ces informations sont mémorisées pour chaque élément dans le graphe d'éléments fonctionnels et ensuite dans l'arbre pour les obtenir dans une chaîne de cotes.
- *Zone droite* : cette zone sert à visualiser le mécanisme.

D'autres commandes ont été prévues dans l'objectif d'une amélioration ultérieure. Par exemple l'identification des éléments fonctionnels par des couleurs différentes.

#### 4.2.2.2. L'option : Sélectionner d'une condition fonctionnelle

Cette option permet de choisir une condition fonctionnelle entre deux pièces différentes. Pour faire le choix, on utilise le même principe que les sélections de contacts entre deux éléments fonctionnels.

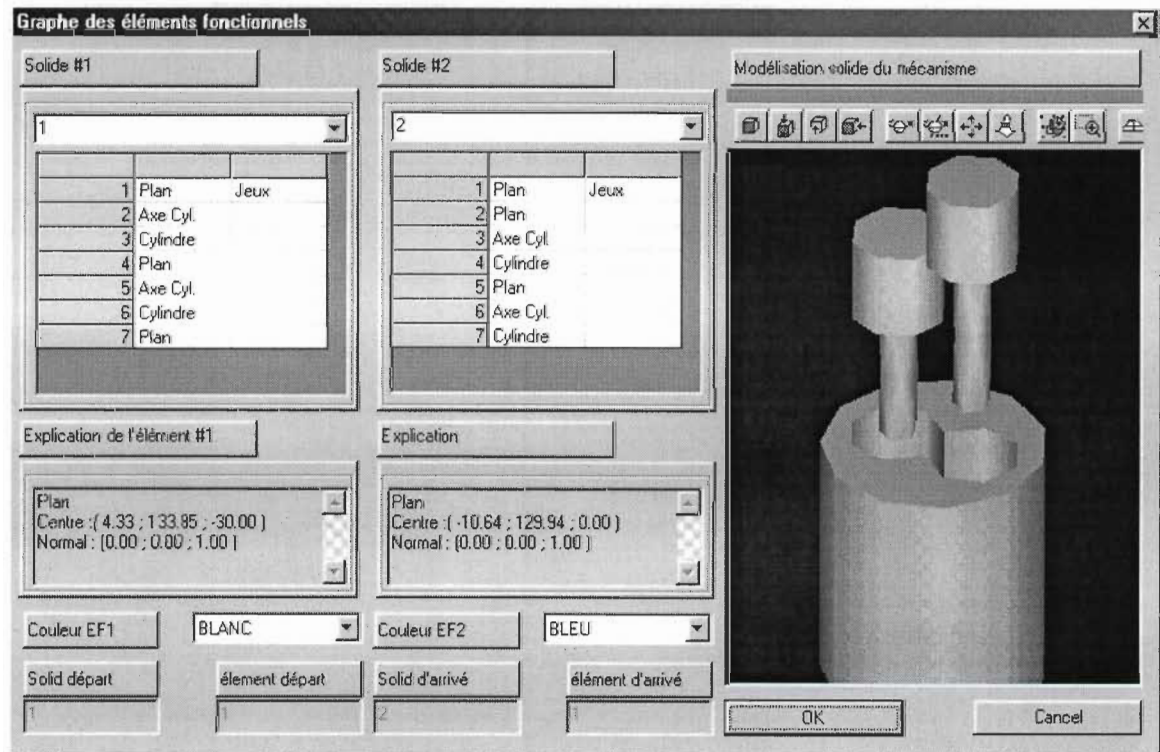


Figure 4.10 : Écran de sélection d'une condition fonctionnelle.

L'organisation de la fenêtre de choix d'une condition fonctionnelle est la même celle de la fenêtre de choix des contacts.



#### 4.2.2.3. L'option : Générer le graphe d'éléments fonctionnels

Une fois que les éléments fonctionnels en contact et la condition fonctionnelle sont sélectionnés, nous pouvons établir le graphe d'éléments fonctionnels à partir de l'option <Générer graphe> d'éléments fonctionnels à partir du menu principal.

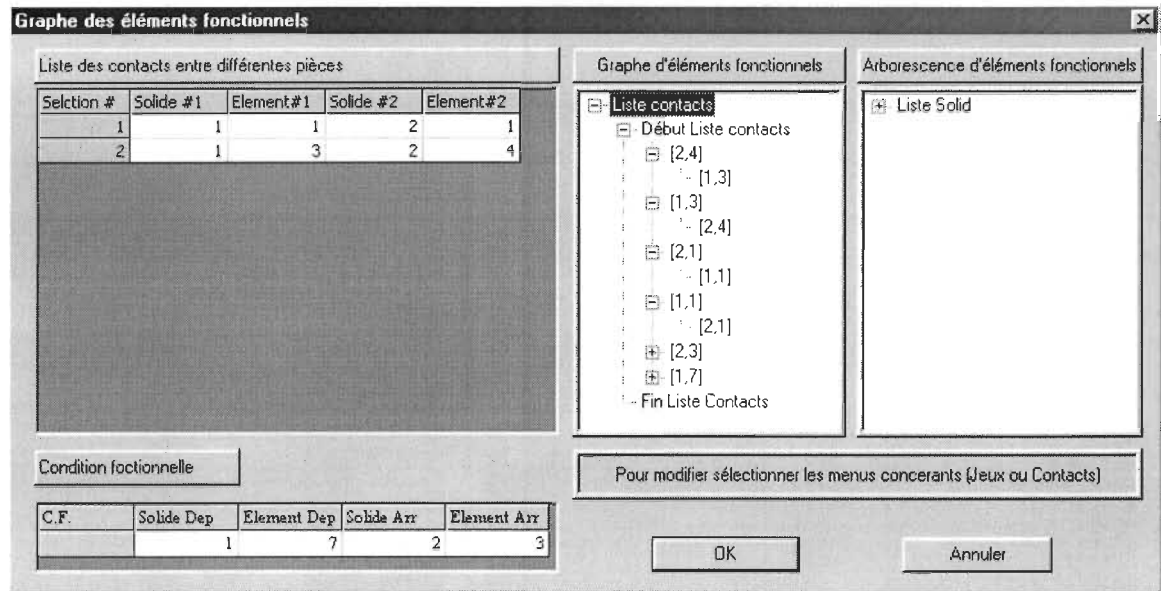


Figure 4.11 : Écran de graphe d'éléments fonctionnels

La fenêtre graphe d'éléments fonctionnels est divisée en quatre parties :

1. Liste des contacts entre différentes pièces ;
2. La condition fonctionnelle à respecter ;
3. Le graphe d'éléments fonctionnels généré automatiquement par le programme ;
4. L'arborescence d'éléments fonctionnels du mécanisme établie automatiquement.

Si l'option <**OK**> est choisie, le graphe sera alors approuvé pour ensuite choisir l'option <**Construction arbre**> pour obtenir l'arbre des éléments fonctionnels.

### 4.2.3. Menu : Arbre des paires d'éléments

#### 4.2.3.1 L'option : Construction Arbre

Cette option permet de construire et visualiser l'arbre des éléments fonctionnels.

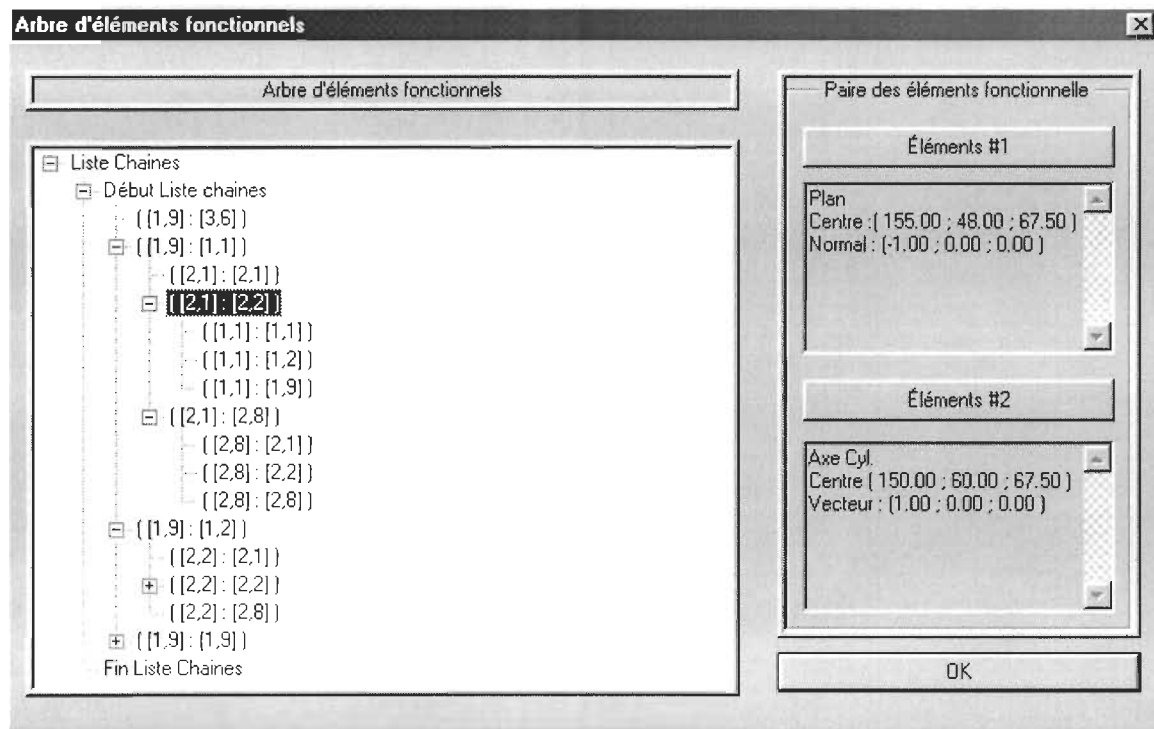


Figure 4.12 : Écran d'arbre d'éléments fonctionnels

La fenêtre 4.12 est divisée en deux parties :

1. L'arbre des éléments fonctionnels au complet; il faut tout simplement cliquer sur le signe (+) pour avoir la chaîne complète.
2. La position et l'orientation d'une paire d'éléments fonctionnels de l'arbre.

#### 4.2.3.2. L'option : Résultats des chaînes de cotes

Cette option permet d'obtenir la liste des chaînes de cotes générées automatiquement par le programme. (Figure 4.13)

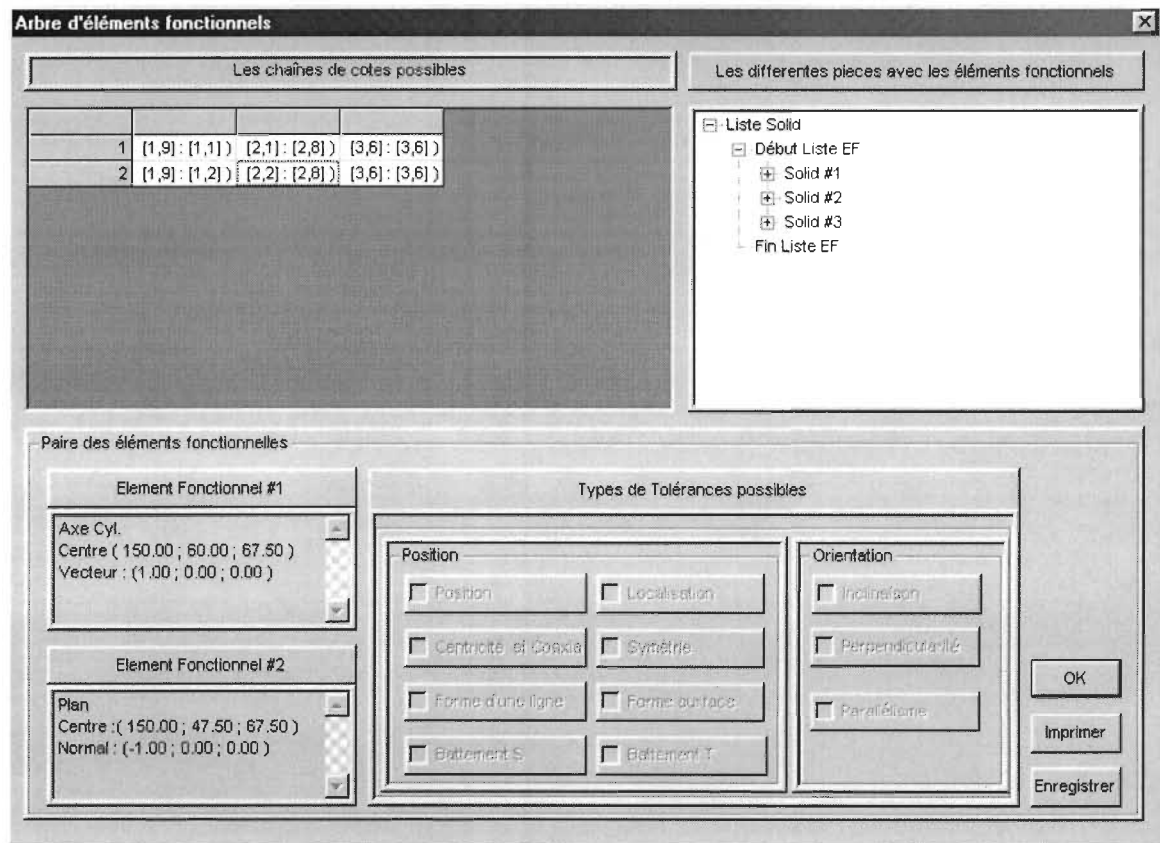


Figure 4.13 : Écran résultats des chaînes de cotes

Cette fenêtre contient un tableau des chaînes de cotes. Chaque ligne de ce tableau est une chaîne. Les choix <Imprimer> et <Enregistrer> s'appliquent seulement sur le tableau des chaînes de cotes

#### Notes :

- ◆ Chaque case est une paire d'éléments fonctionnels et le deuxième élément fait contact avec le premier élément de la case suivante
- ◆ Le premier élément de la première case d'une ligne est l'élément départ de la condition fonctionnelle et le dernier élément dans la ligne est l'élément d'arrivée dans la C.F.
- ◆ Dans la zone de droite en bas se trouvent les possibilités de tolérances à appliquer sur une paire d'éléments : il suffit tout simplement de cliquer sur la case correspondante dans le tableau de chaînes de cotes

#### 4.2.3.3. L'option : Détail des chaînes de cotes

Cette option permet d'obtenir tous les détails de la génération des chaînes de cote autour de la condition fonctionnelle.

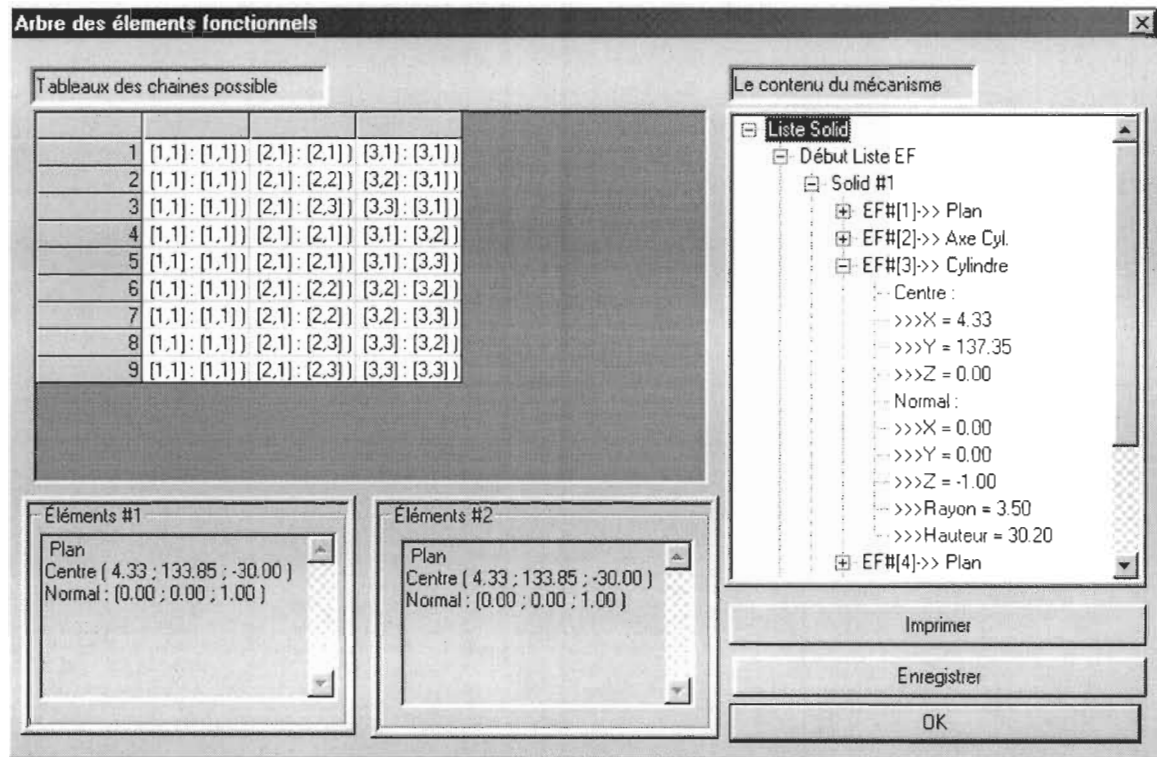


Figure 4.14 : Écran détail de la liste de chaîne des cotes

## CHAPITRE 5

### EXEMPLE D'APPLICATION

---

#### ***5.1. APPLICATION MANUELLE***

- 5.1.1. Représentation du mécanisme à analyser
- 5.1.2. Éléments fonctionnels
- 5.1.3. Graphe d'éléments fonctionnels
- 5.1.4. Chaînes de cotes

#### ***5.2. CHEMINS POSSIBLES***

#### ***5.3. RÉSULTATS DU PROGRAMME***

- 5.3.1. Graphe d'éléments fonctionnels
- 5.3.2. Chaînes de cotes automatiques
- 5.3.3. Comparaison des différents résultats

Pour valider notre algorithme nous allons procéder en trois étapes sur un exemple afin d'obtenir toutes les chaînes de cotes possibles autour d'une condition imposée (CF).

Les trois étapes sont :

***1. Exécuter les étapes de l'algorithme manuellement :***

Nous allons en premier lieu trouver manuellement tous les éléments fonctionnels du mécanisme. Ensuite nous trouverons tous les contacts entre différentes pièces et nous imposerons une condition fonctionnelle pour bâtir le graphe des éléments fonctionnels afin d'obtenir toutes les chaînes possibles autour de la condition fonctionnelle imposée. Ensuite exécuter le programme (Chemins possibles) pour avoir une idée sur les chaînes obtenues.

***2. Exécuter le programme en utilisant l'exemple dessiné sur AUTOCAD 14 :***

Cette étape consiste tout simplement à charger le fichier dessin en 3D sur AUTOCAD 14 et sélectionner les contacts et la condition fonctionnelle. Le programme devra fournir toutes les chaînes possibles autour de cette condition.

***3. Comparer les deux résultats obtenus aux étapes précédentes.***

L'exemple que nous avons choisi, pour valider notre algorithme, est un mécanisme constitué de trois pièces. Ce mécanisme est une pompe à engrenages que nous avons simplifié. Cette pompe est constituée de trois pièces : le corps principal de la pompe (figure 5.2) et les deux arbres (figure 5.1).

À l'aide de la définition fonctionnelle des pièces 1, 2 et 3 nous avons défini une condition fonctionnelle entre les deux axes des arbres.

## 5.1. APPLICATION MANUELLE

### 5.1.1 Représentation du mécanisme à analyser

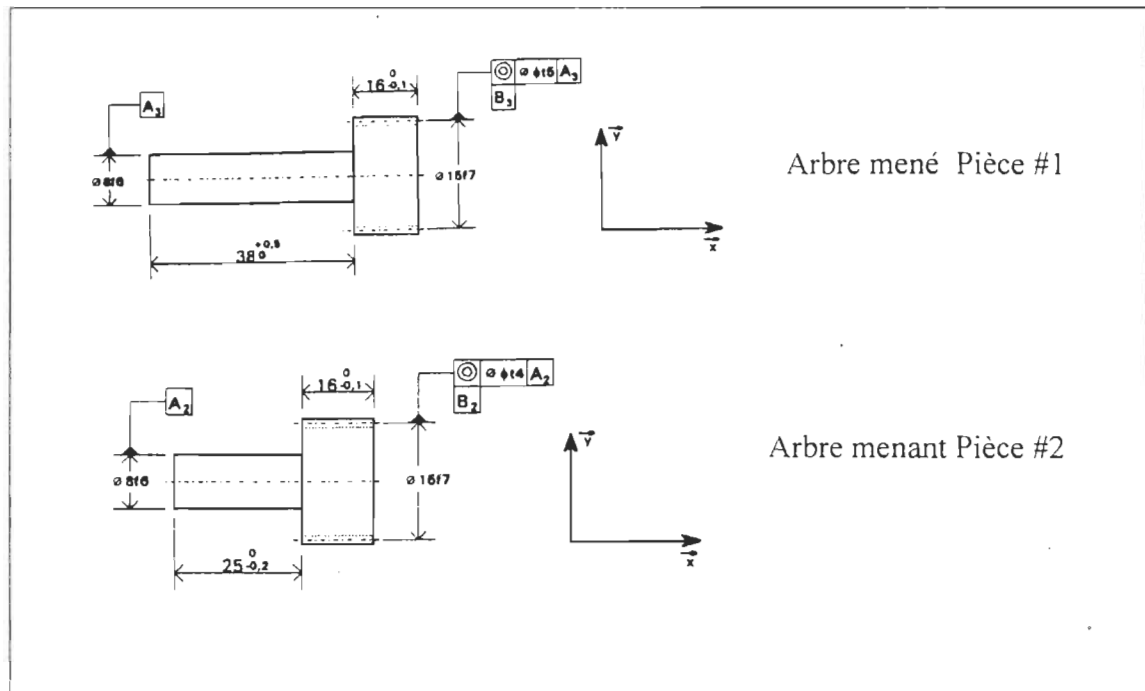


Figure 5.1 : Mécanisme de la pompe à engrenage : arbre mené et arbre menant

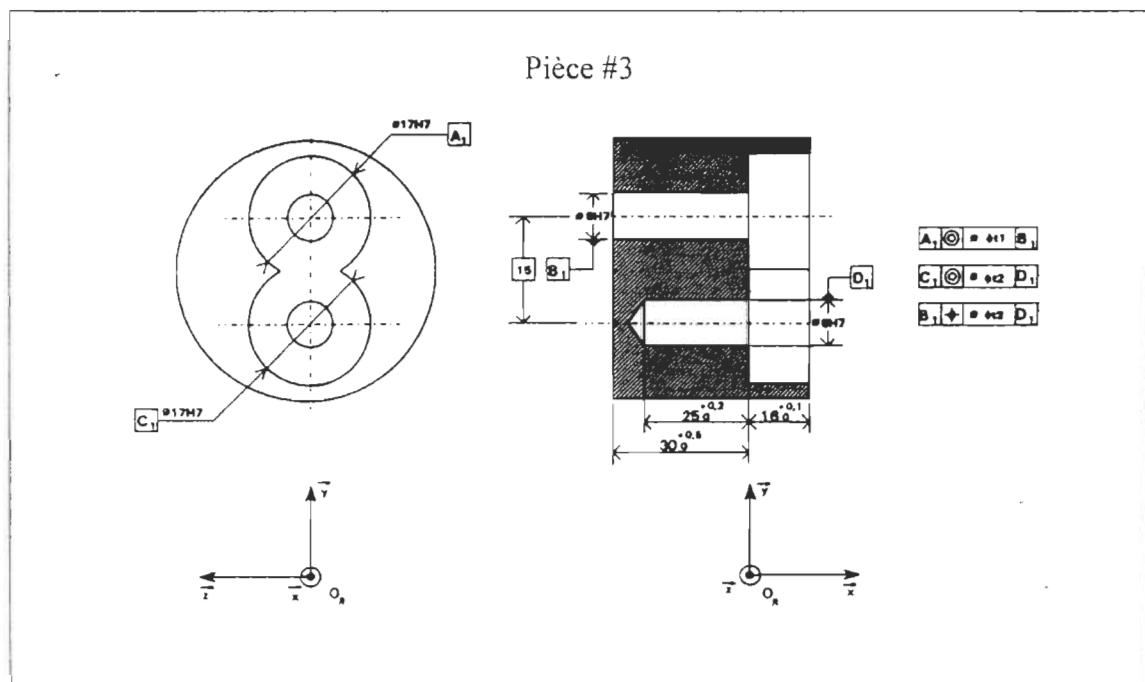


Figure 5.2 : Mécanisme de la pompe à engrenage : corps principal

### 5.1.2. Éléments fonctionnels

La première étape dans l'algorithme consiste à trouver tous les éléments fonctionnels dans le mécanisme en question. Dans cette section on se préoccupe de déterminer manuellement la liste de ces éléments fonctionnels dans le mécanisme de la pompe à engrenages.

La figure 5.3 montre les éléments fonctionnels de la pièce #1 et #2.

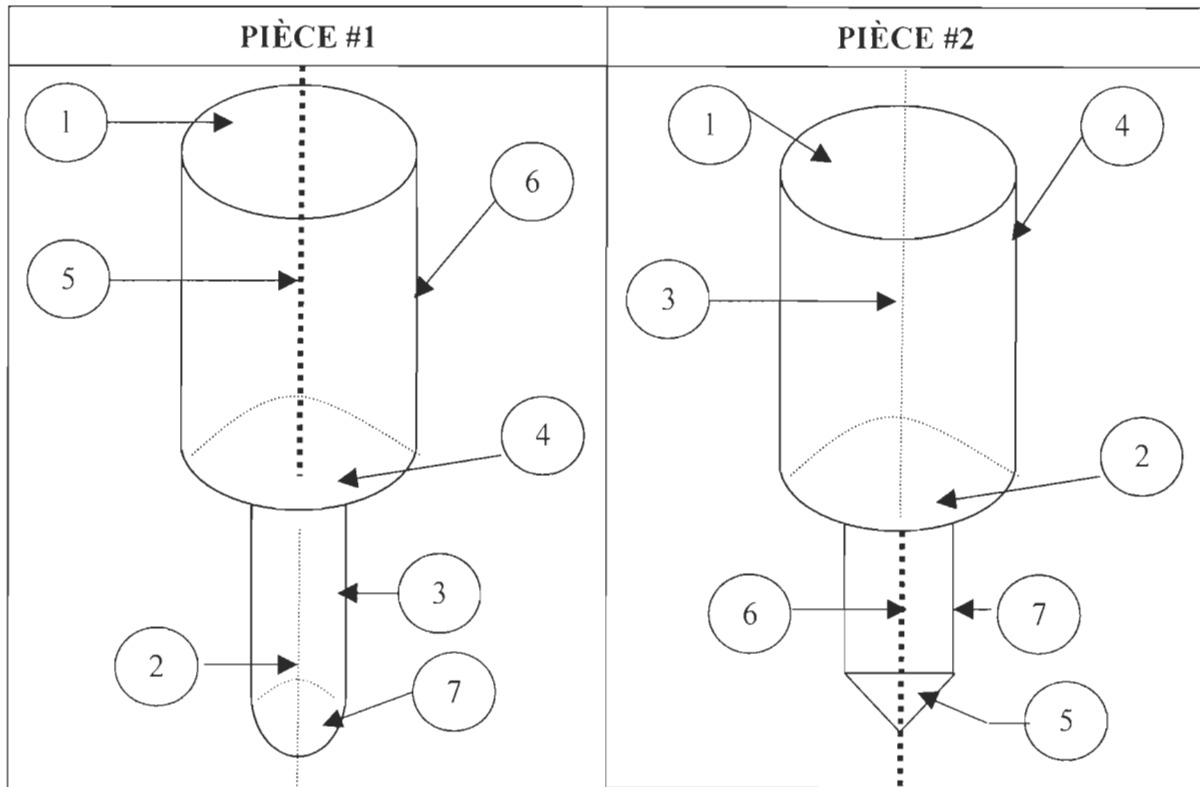


Figure 5.3 Éléments fonctionnels de la pièce #1 et #2 du mécanisme pompe à engrenage

Les éléments fonctionnels sont :

Pièce #1 = {1, 2, 3, 4, 5, 6, 7}

Pièce #2 = {1, 2, 3, 4, 5, 6, 7}

Afin de valider l'algorithme nous imposons une condition fonctionnelle entre les deux axes de deux arbres (pièce #1 et #2). La condition fonctionnelle devient {[1,5], [2,6]}



La pièce #3 contient 14 éléments fonctionnels, pour faciliter la visualisation de ces éléments la pièce est dessiner en deux parties. La partie **extérieure** à gauche montre les éléments fonctionnels externes de la pièce et la partie **intérieure** droite montre les éléments intérieurs de la pièce.

La figure 5.4 montre les éléments fonctionnels de la pièce #3

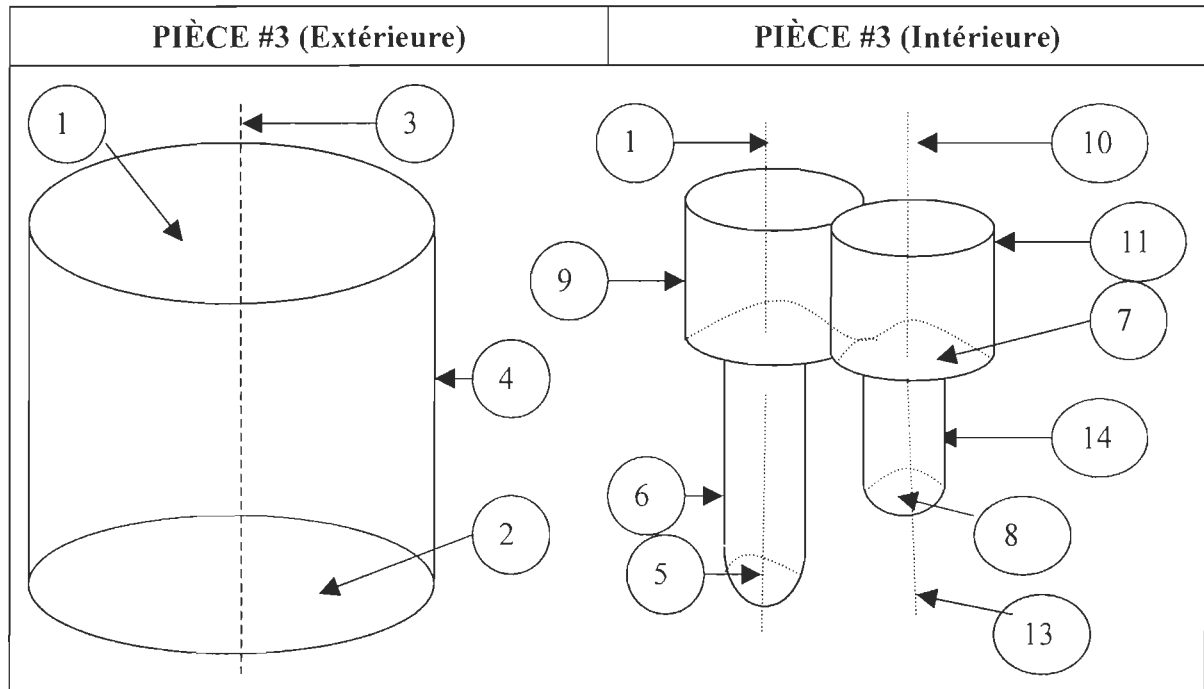


Figure 5.4 Éléments fonctionnels de la pièce #3 du mécanisme pompe à engrenage

Éléments fonctionnels de la pièce #3 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}

### 5.1.3 Graphe d'éléments fonctionnels

Nous utilisons les éléments fonctionnels obtenus auparavant pour déterminer la liste totale des éléments fonctionnels L et nous pouvons déterminer aussi les éléments en contacts entre différentes pièces et la condition fonctionnelle à respecter.

La liste des éléments fonctionnels est :

$$L = \{\{1, 2, 3, \dots, 7\}, \{1, 2, 3, \dots, 7\}, \{1, 2, 3, \dots, 14\}\}.$$

À noter ici que cette liste contient tous les éléments existants dans le mécanisme non seulement les éléments des contacts. Cette liste sert à sélectionner les contacts et la condition fonctionnelle.

Nous pouvons procéder à la détermination des contacts en utilisant la liste des éléments fonctionnels et le dessin à 3D. La liste des contacts est la suivante :

$$C1 = \{[1,3],[3,6]\}, C2 = \{[1,4],[3,7]\}, C3 = \{[1,6],[3,9]\}, C4 = \{[3,7],[2,2]\}$$

$$C5 = \{[3,8],[2,5]\}, C6 = \{[3,11],[2,4]\}, C7 = \{[3,14],[2,7]\}.$$

$$C = \{ C1, C2, C3, C4, C5, C6, C7 \}$$

La condition fonctionnelle à respecter imposée plus haut est :

$$J = \{ ([1,5],[2,6]) \}$$

Utilisons à présent la liste des éléments L, la liste des contacts C et la condition fonctionnelle J pour déterminer le graphe d'éléments fonctionnels initial  $G = \{L_G, R_G, \psi_G\}$ . La figure 5.5 montre la représentation graphique du graphe G.

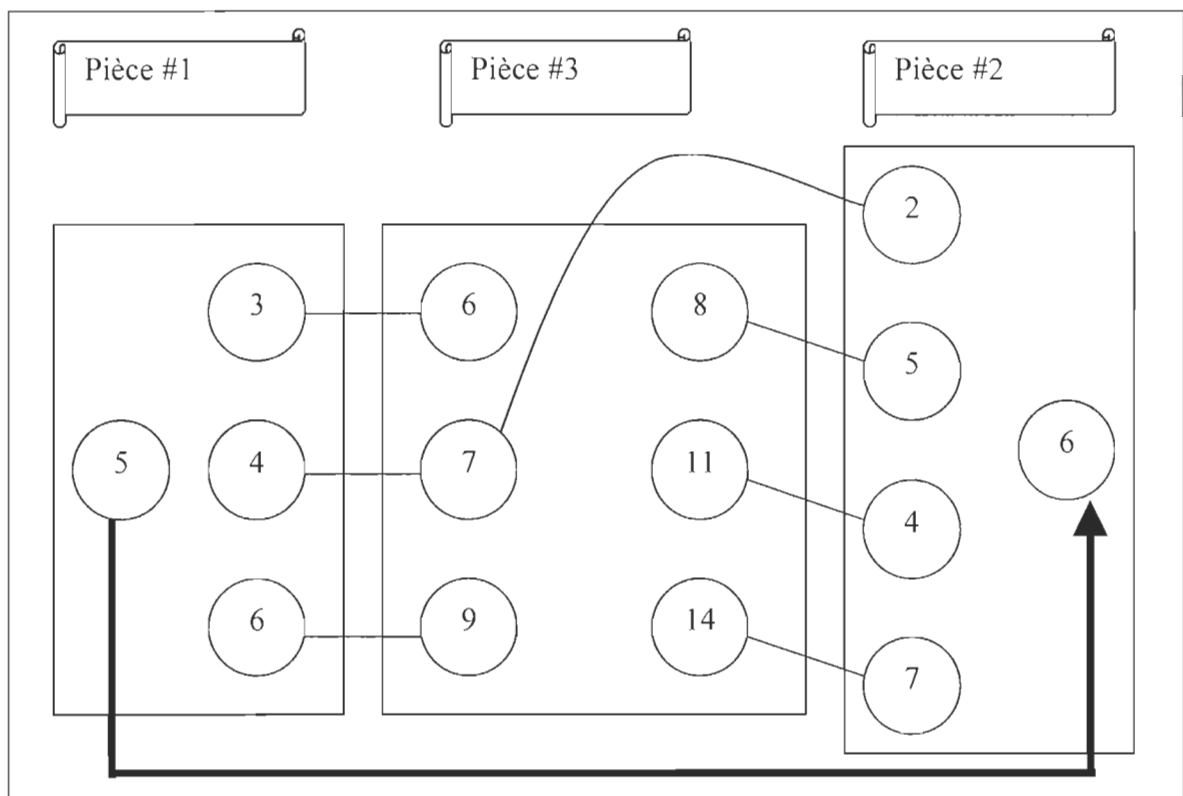


Figure 5.5 : Graphe des éléments fonctionnels initial pour le mécanisme de la pompe.

Nous pouvons remarquer ici que la connexion dans le graphe d'éléments fonctionnels jusqu'à maintenant est seulement au niveau des contacts, les paires d'éléments n'apparaissent pas présentement.

Pour obtenir le graphe au complet il faut obtenir sa matrice incidence. L'obtention de cette matrice sera effectuée par itération qui est basée sur la détermination des paires d'éléments fonctionnels entre différents éléments appartenant à une même pièce.

Nous pouvons procéder maintenant à construire la matrice d'incidence en fonction du graphe obtenu à la figure 5.18. Premièrement nous allons placer les contacts ( $C_i$ ) à leurs positions concernant dans la matrice pour ensuite trouver les éléments qui peuvent former des paires d'éléments dans une même pièce.

$C1 = \{[1,3], [3,6]\}$  est un contact entre les deux éléments  $[1,3]$  et  $[3,6]$  qui apparaît deux fois dans la matrice parce que par définition les contacts sont des fonctions symétriques. Ce principe est appliqué sur tous les contacts dans la matrice d'incidence (Tableau 5.1).

Tableau 5.1 Matrice d'incidence initiale du graphe d'éléments fonctionnels

L X L		Pièce #1				Pièce #3						Pièce #2				
		3	4	5	6	6	7	8	9	11	14	2	4	5	6	7
Pièce #1	3					C1										
	4						C2									
	5															
	6								C3							
Pièce #3	6	C1														
	7		C2									C4				
	8													C5		
	9				C3											
	11												C6			
	14															C7
Pièce #2	2						C4									
	4									C6						
	5							C5								
	6															
	7										C7					

Pour compléter le reste de la matrice d'incidence il faut déterminer les paires d'éléments fonctionnels dans chacune de pièce et ce-ci peut se faire par itération dans la matrice comme mentionner auparavant. L'itération commence par l'élément de départ de la condition fonctionnelle et se termine par l'élément d'arrivée de celle-ci.

Pour démarrer le processus de l'itération nous commençons par trouver les éléments fonctionnels qui peuvent former une paire avec l'élément de départ de la condition fonctionnelle. Ce-ci peut se faire en prenant tous les éléments horizontaux dans la même pièce qu'elle appartienne, pour différencier le contact d'une paire nous allons placer **P** (Paire d'éléments fonctionnels) dans la case qui peuvent former une paire avec la condition fonctionnelle. (Tableau 5.2 cases de la pièce #1)

L'élément de départ de la condition fonctionnelle est  $[1,5]$  alors il faut ajouter **P** dans chacune des cases de la ligne où se trouve l'élément 5 de la pièce #1. Remarquons que les éléments  $\{[1,3], [1,4], [1,6]\}$  peuvent former paire avec l'élément  $[1,5]$ . Par contre l'élément  $[1,5]$  n'est pas

dans la liste car un élément ne peut former une paire avec lui-même sauf dans le cas où il n'y a qu'un seul contact et un seul élément.

Tableau 5.2 Matrice d'incidence intermédiaire du graphe d'éléments fonctionnels

L X L		Pièce #1				Pièce #3						Pièce #2				
		3	4	5	6	6	7	8	9	11	14	2	4	5	6	7
Pièce #1	3					C1										
	4						C2									
	5	P	P		P											
	6								C3							
Pièce #3	6	C1					P	P	P	P	P					
	7		C2									C4				
	8													C5		
	9				C3											
	11												C6			
	14															C7
Pièce #2	2						C4								P	
	4									C6						
	5							C5								
	6															
	7										C7					

Une fois les paires d'éléments fonctionnels sont identifiées dans la première pièce nous pouvons procéder pour chacune de paire à trouver ses contacts ceci peut se faire en prenant tous les contacts qui se trouvent verticalement dans la même colonne où se trouve la lettre P pour recommencer le même principe de départ (trouver toutes les paires dans même pièce horizontalement). Une condition importante à vérifier : une pièce ne peut être sélectionner qu'une fois pendant le remplissage de la matrice. Après les parcours de toutes les pièces nous aurons la matrice d'incidence complète (Tableau 5.3).

Remarques :

- 1 La première pièce est toujours la pièce où se trouve l'élément de départ de la condition fonctionnelle et la dernière pièce est la pièce où se trouve l'élément d'arrivé de celle-ci.
- 2 Arrivant à la dernière pièce un élément ne peut former une paire seulement avec l'élément d'arrivé de la condition fonctionnelle. Exemple dans le (tableau 5.3 ) l'élément [3,6] a un contact avec [2,2] et ce dernier ne peut faire paire qu'avec l'élément [2,6].

Tableau 5.3 Matrice d'incidence complète du graphe d'éléments fonctionnels

L X L		Pièce #1				Pièce #3						Pièce #2				
		3	4	5	6	6	7	8	9	11	14	2	4	5	6	7
Pièce #1	3					C1										
	4						C2									
	5	P	P		P											
	6								C3							
Pièce #3	6	C1					P	P	P	P	P					
	7		C2			P		P	P	P	P	C4				
	8					P	P		P	P	P			C5		
	9				C3	P	P	P		P	P					
	11					P	P	P	P		P		C6			
	14					P	P	P	P	P						C7
Pièce #2	2						C4								P	
	4									C6					P	
	5							C5							P	
	6															
	7										C7				P	

À partir de la matrice d'incidence nous pouvons produire le graphe d'éléments fonctionnels pour produire ensuite l'arbre d'éléments fonctionnels. (Figure 5.6)

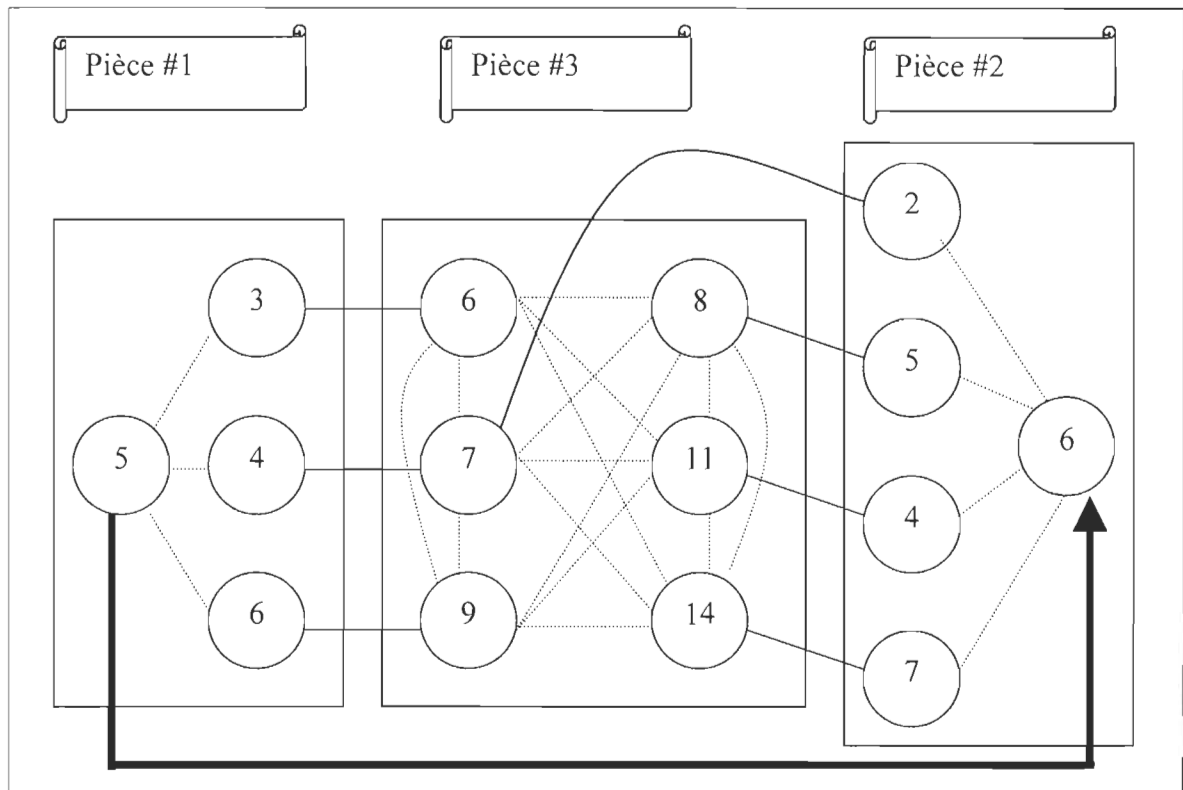


Figure 5.6 : Graphe des éléments fonctionnels complet pour le mécanisme de la pompe.

### 5.1.4 Chaînes de cotes

Le Graphe d'éléments fonctionnels peut être présenté par la matrice d'incidence complète (Tableau 5.3) ou par sa représentation graphique (figure 5.6). le parcours de ce dernier est possible par les différentes méthodes du parcours de graphe classique en fixant la tête à l'élément de départ de la condition fonctionnelle tandis que la fin à l'élément d'arrivée de la C.F.

En utilisant la matrice d'incidence (tableau 5.3) ou le GEF de la figure 5,6 nous pouvons trouver toutes les chaînes autour de la condition fonctionnelle  $\{[1,5][2,6]\}$ :

1.  $\rightarrow ([1,5] : [1,3]) \rightarrow ([3,6] : [3,7]) \rightarrow ([2,2] : [2,6])$
2.  $\rightarrow ([1,5] : [1,3]) \rightarrow ([3,6] : [3,8]) \rightarrow ([2,5] : [2,6])$
3.  $\rightarrow ([1,5] : [1,3]) \rightarrow ([3,6] : [3,11]) \rightarrow ([2,4] : [2,6])$
4.  $\rightarrow ([1,5] : [1,3]) \rightarrow ([3,6] : [3,14]) \rightarrow ([2,7] : [2,6])$
5.  $\rightarrow ([1,5] : [1,4]) \rightarrow ([3,7] : [3,8]) \rightarrow ([2,5] : [2,6])$
6.  $\rightarrow ([1,5] : [1,4]) \rightarrow ([3,7] : [3,11]) \rightarrow ([2,4] : [2,6])$
7.  $\rightarrow ([1,5] : [1,4]) \rightarrow ([3,7] : [3,14]) \rightarrow ([2,7] : [2,6])$
8.  $\rightarrow ([1,5] : [1,6]) \rightarrow ([3,9] : [3,7]) \rightarrow ([2,2] : [2,6])$
9.  $\rightarrow ([1,5] : [1,6]) \rightarrow ([3,9] : [3,8]) \rightarrow ([2,5] : [2,6])$
10.  $\rightarrow ([1,5] : [1,6]) \rightarrow ([3,9] : [3,11]) \rightarrow ([2,4] : [2,6])$
11.  $\rightarrow ([1,5] : [1,6]) \rightarrow ([3,9] : [3,14]) \rightarrow ([2,7] : [2,6])$

On peut remarquer que seulement 11 chemins possibles peuvent être tracer autour de la condition fonctionnelle  $([1,5] : [2,6])$ .

## 5.2. CHEMINS POSSIBLES

Afin de valider le nombre de chemins trouvés nous avons exécuter le programme (Chemins Possibles). Après avoir introduit la matrice d'incidence trouvée dans la section précédente et la longueur de la chaîne demandée le programme nous fournit l'écran suivant :

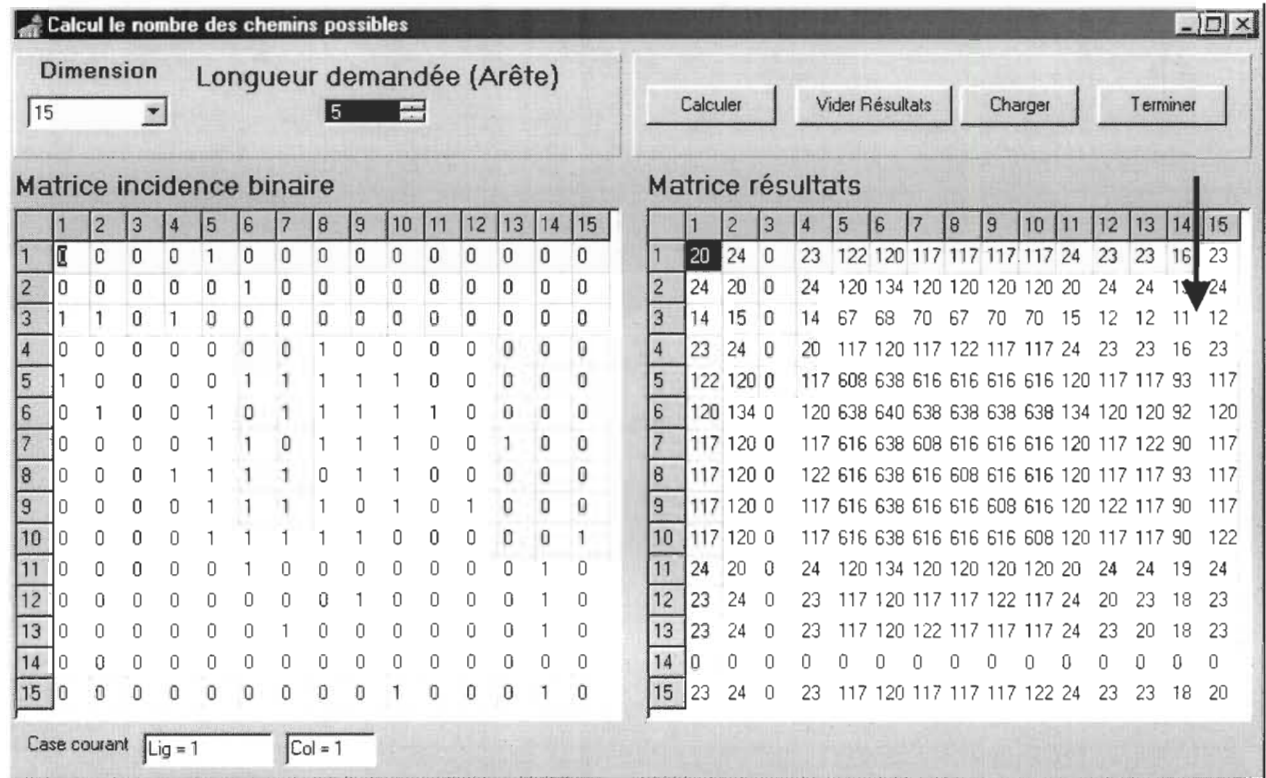


Figure 5.7 : Résultats de tous les chemins possibles pour pompe à engrenage

Nous pouvons remarquer qu'il y a **11** chemins possibles entre les points 3 et 14 qui correspondent respectivement à [1,3] et [2,6]. ([1,3] et [2,6] est La condition fonctionnelle à respecter ).

Notes : le nombres de chemins obtenus par ce programme donne une idée sur les nombres de chaînes obtenues manuellement.

## 5.3. Résultats du programme

### 5.3.1 Graphe d'éléments fonctionnels

Dans cette étape il faut tout simplement introduire le fichier solide de l'exemple au programme (Analyse en TAO). Le programme détecte automatiquement tous les éléments fonctionnels. Ensuite nous allons sélectionner les éléments en contacts et la condition fonctionnelle entre différentes pièces.

Pour générer le graphe des éléments fonctionnels nous avons lancé l'option **<Générer graphe>** à partir du menu principal pour obtenir la fenêtre (figure 5.20).

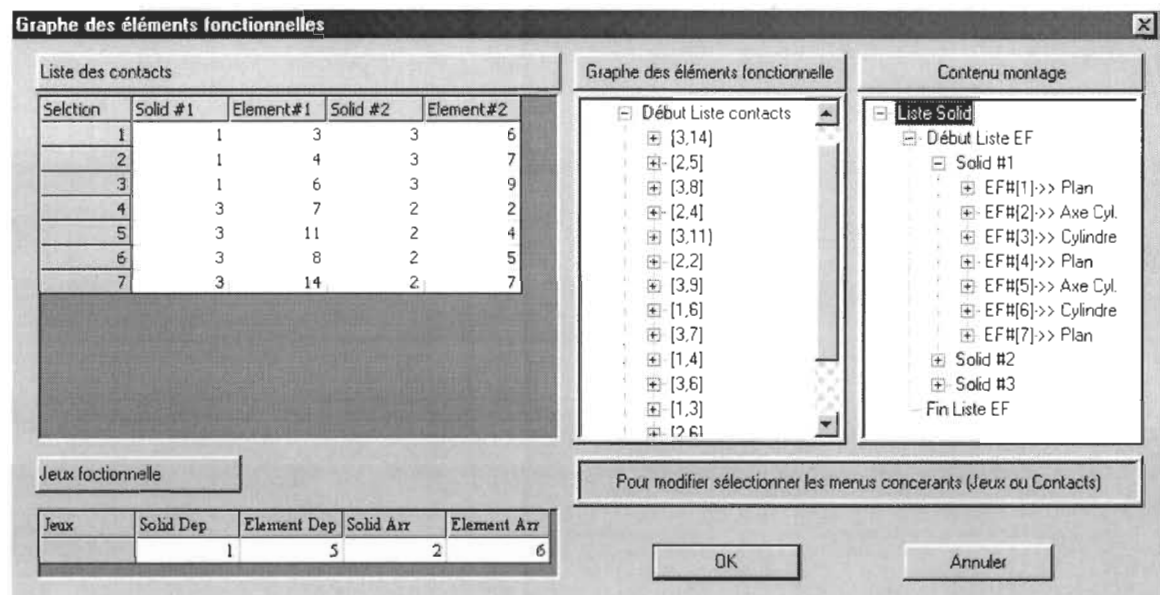


Figure 5.8 : Écran graphe d'éléments fonctionnels de l'exemple pompe à engrenage

Après l'obtention du graphe nous avons lancé l'option **<Construction arbre>** à partir du menu principal pour obtenir l'arbre des éléments fonctionnels.



### 5.3.2 Chaînes de cotes automatiques

La figure 5.21 montre toutes les chaînes de cotes générées automatiquement par le programme après avoir lancé l'option **<Résultats des chaînes de cotes>** à partir du menu principal.

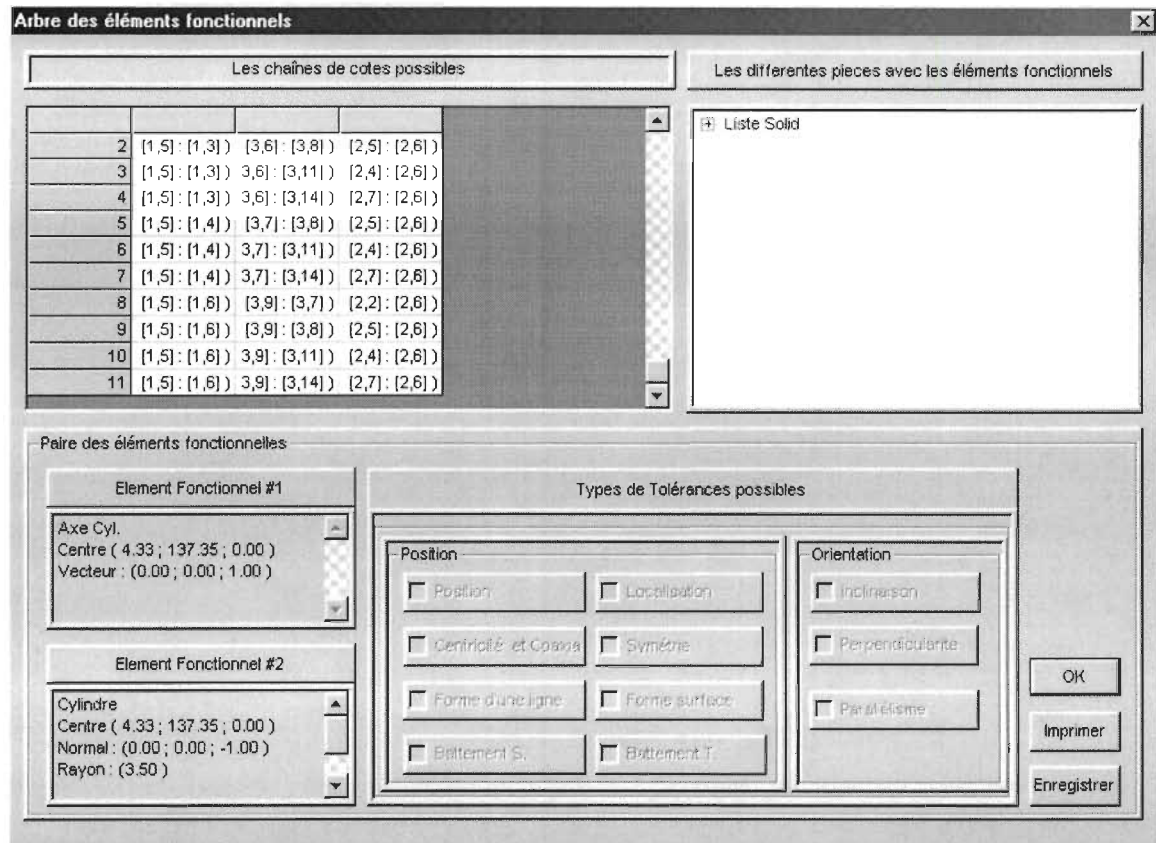


Figure 5.9 : Écran des chaînes de cotes de l'exemple pompe à engrenage

Les chaînes de cotes générées sont enregistrées dans un fichier montré au tableau 5.2 à la page suivante.

### 5.3.3 Comparaisons des différents résultats

Nous pouvons conclure que les résultats obtenus manuellement et par le programme (Analyse en TAO) sont conformes et validés par le programme (Chemins possibles) qui a déterminé qu'il y avait 11 possibilités de chaînes de cotes.

Le tableau 5.4 montre le fichier résultat généré par le programme

Tableau 5.4 : Fichier résultat généré par le programme Analyse TAO

---

Analyse fonctionnelle  
Fichier <Pump.sat >

---

Les contacts sélectionnés :

Sel#	S#1	E#1	S#2	E#2
1	1	3	3	6
2	1	4	3	7
3	1	6	3	9
4	3	7	2	2
5	3	11	2	4
6	3	8	2	5
7	3	14	2	7

---

Condition fonctionnelle à respecter :

Solide départ : 1      Élément départ : 5  
Solide d'arrivée : 2    Élément d'arrivée : 6

---

Résultats de chaînes de cotes tridimensionnelles

1 ->	( [1,5] : [1,3] )->( [3,6] : [3,7] )->( [2,2] : [2,6] )->
2 ->	( [1,5] : [1,3] )->( [3,6] : [3,8] )->( [2,5] : [2,6] )->
3 ->	( [1,5] : [1,3] )->( [3,6] : [3,11] )->( [2,4] : [2,6] )->
4 ->	( [1,5] : [1,3] )->( [3,6] : [3,14] )->( [2,7] : [2,6] )->
5 ->	( [1,5] : [1,4] )->( [3,7] : [3,8] )->( [2,5] : [2,6] )->
6 ->	( [1,5] : [1,4] )->( [3,7] : [3,11] )->( [2,4] : [2,6] )->
7 ->	( [1,5] : [1,4] )->( [3,7] : [3,14] )->( [2,7] : [2,6] )->
8 ->	( [1,5] : [1,6] )->( [3,9] : [3,7] )->( [2,2] : [2,6] )->
9 ->	( [1,5] : [1,6] )->( [3,9] : [3,8] )->( [2,5] : [2,6] )->
10 ->	( [1,5] : [1,6] )->( [3,9] : [3,11] )->( [2,4] : [2,6] )->
11 ->	( [1,5] : [1,6] )->( [3,9] : [3,14] )->( [2,7] : [2,6] )->

---

Fin Fichier

## CHAPITRE 6

### CONCLUSION

---

#### 6.1 ACCOMPLISSEMENTS

- 6.1.1 Identification des éléments fonctionnels
- 6.1.2 Génération du graphe des éléments fonctionnels
- 6.1.3 Identification des cycles autour d'une condition fonctionnelle
- 6.1.4 Interface graphique utilisateur

#### 6.2 UTILISATIONS POSSIBLES

- 6.2.1 Détermination des équations de l'intervalle de tolérance
- 6.2.2 Analyse de l'influence de chaque chaîne
- 6.2.3 Modélisation mathématique
- 6.2.4 Analyse de robustesse du design
  - 6.2.4.1 Axiomes de robustesse
  - 6.2.4.2 Application au tolérancement

#### 6.3 LIMITES ET RECOMMANDATIONS

À travers cette recherche, nous avons constaté que le domaine du tolérancement assisté par ordinateur a fait beaucoup de progrès depuis plusieurs années. Ces recherches ont été orientées vers l'élaboration des algorithmes ainsi que des méthodes de représentation géométrique afin de faciliter l'analyse fonctionnelle d'un mécanisme en tolérancement assisté par ordinateur. Malgré tout, beaucoup reste à faire.

Dans ce chapitre, nous allons discuter des accomplissements réalisés par notre étude et ensuite nous allons présenter les limites de l'algorithme. Nous terminerons par des recommandations pour les recherches futures.

## **6.1 ACCOMPLISSEMENTS**

L'étude proposée a permis de mettre en évidence et de valider un modèle de tolérancement assisté par ordinateur. Ce modèle consiste à bâtir le graphe des éléments fonctionnels, après avoir déterminé la liste des éléments fonctionnels d'un mécanisme modélisé par un modèleur 3D et sélectionner les contacts possibles entre différentes pièces. Une fois que le graphe des éléments fonctionnels a été établi, il suffit tout simplement de déterminer la condition fonctionnelle à respecter pour le modèle en question, et ensuite déterminer toutes les chaînes de cotes autour de cette condition.

### **6.1.1 Identification des éléments fonctionnels**

Les éléments fonctionnels utilisés à travers cette recherche sont toutes les formes géométriques fictives (Ex. Axe) ou réels (Ex. Plan) qui agissent sur la fonctionnalité de la pièce comme des éléments fonctionnels.

Notre algorithme nous permet l'extraction des éléments fonctionnels d'un dessin technique à partir d'un modèleur solide 3D enregistré dans un fichier «sat». L'extraction de ces éléments est obligatoire dans le but d'établir le graphe des éléments fonctionnels pour ensuite obtenir l'arbre des paires de ces éléments.

Les éléments fonctionnels extraits sont mémorisés dans une structure de données facile à utiliser. Cette structure est implantée en utilisant les principes de l'orienté objet en C++, cette dernière contenant toutes les informations et les propriétés des éléments fonctionnels (Plan, cylindre, axe, cône, etc.). Les propriétés mémorisées sont l'orientation et la position avec des données géométriques (les propriétés d'un plan sont : la normale et le centre, pour un cylindre en plus de la normale nous avons le rayon et la hauteur). En tout temps cette structure garde le lien avec le fichier initial «sat» dans un but de continuité avec d'autres travaux ultérieurs. L'impression et l'enregistrement de la nouvelle structure sont possibles, étant donné la complexité du fichier BREP, pour une ultérieure consultation.

### **6.1.2 Génération du graphe des éléments fonctionnels**

Une fois que les éléments fonctionnels ont été déterminés, l'algorithme d'identification des chaînes de cote nous permet d'établir le graphe des éléments fonctionnels et ce après

la sélection des contacts entre les différentes pièces. Les contacts sont sélectionnés manuellement afin de valider notre algorithme.

Le graphe ainsi obtenu contient seulement les éléments en contact, autrement dit nous gardons en mémoire dans le graphe uniquement les éléments en contact. Quant aux autres éléments, ils sont disponibles dans une liste arborescente en gardant le lien avec le fichier BREP initial pour toute ultérieure consultation.

### **6.1.3 Identification des cycles autour d'une condition fonctionnelle**

L'algorithme d'identification des chaînes de cotes utilise le graphe des éléments fonctionnels pour obtenir des cycles autour d'une condition fonctionnelle sélectionnée par l'utilisateur.

Les cycles obtenus sont des chemins possibles dans le graphe des éléments fonctionnels. Ils débutent et se terminent successivement par le premier et le dernier élément de la condition fonctionnelle. Les cycles constituent des paires d'éléments fonctionnels dans une même pièce.

Ces cycles sont mémorisés dans une liste arborescente pour établir l'arbre d'éléments fonctionnels.

### **6.1.4 Interface graphique d'utilisateur**

Une contribution importante dans ce travail est celle la conception d'une interface utilisateur graphique qui permet de saisir des contacts et une condition fonctionnelle afin d'obtenir automatiquement les cycles autour de celle-ci.

Nous offrons une interface graphique avec un menu déroulant à trois options (Fichier, Graphe des éléments fonctionnels et Arbre des paires d'éléments fonctionnels) et chaque option contient un sous menu.

L'option Fichier sert à charger et visualiser un dessin modélisé sur un modèleur 3D. Le rôle de l'option Graphe d'éléments fonctionnels est de saisir les contacts entre différentes pièces et une condition fonctionnelle pour ensuite générer le graphe des éléments fonctionnels. Finalement, l'option Arbre des paires d'éléments fonctionnels sert à parcourir le graphe des éléments fonctionnels en mémoire pour construire l'arbre des paires d'éléments fonctionnels et ainsi obtenir toutes les chaînes de cotes possibles autour de la condition fonctionnelle sélectionnée que l'on doit respecter.

Comme option, l'algorithme fournit une aide à l'utilisateur pour expliquer la fonctionnalité et la base théorique dont on s'est inspiré.

## 6.2 UTILISATIONS POSSIBLES

Le présent algorithme peut être appliqué à plusieurs tâches distinctes : Détermination des équations de l'intervalle de tolérance; identification de la ou des chaînes les plus influentes; quantifications des tolérances et analyse de la robustesse d'un mécanisme.

### 6.2.1 Détermination des équations de l'intervalle de tolérance

À partir de l'arbre des éléments fonctionnels, nous pouvons déduire des équations mathématiques qui satisfont la condition fonctionnelle à respecter.

L'arbre des éléments fonctionnels contient des cycles de cotes. Ces cycles sont constitués des paires d'éléments fonctionnels qui appartiennent alternativement à une même pièce (paire interne) et à des pièces différentes (paire cinématique). Prenant l'exemple de la pompe à engrenages, nous avons 11 chaînes possibles donc nous obtenons 11 équations pour ce mécanisme autour de la condition fonctionnelle à respecter (voir tableau 6.1).

Nous pouvons appliquer l'équation (6.1) pour obtenir toutes les équations tridimensionnelles autour de la condition fonctionnelle. Il suffit seulement d'identifier, dans ces équations, les cotes contenantes et contenues.

$$ITJ = \sum IT(\text{de chaque paires}) \quad 6.1$$

Le tableau 6.1 montre les différentes équations de l'exemple pompe à engrenages.

Tableau 6.1 : Équations d'intervalle de tolérances de l'exemple pompe à engrenages

1.  $IT([1,5] : [1,3]) + IT([3,6] : [3,7]) + IT([2,2] : [2,6]) = IT \text{ C.F.}$
2.  $IT([1,5] : [1,3]) + IT([3,6] : [3,8]) + IT([2,5] : [2,6]) = IT \text{ C.F.}$
3.  $IT([1,5] : [1,3]) + IT([3,6] : [3,11]) + IT([2,4] : [2,6]) = IT \text{ C.F.}$
4.  $IT([1,5] : [1,3]) + IT([3,6] : [3,14]) + IT([2,7] : [2,6]) = IT \text{ C.F.}$
5.  $IT([1,5] : [1,4]) + IT([3,7] : [3,8]) + IT([2,5] : [2,6]) = IT \text{ C.F.}$
6.  $IT([1,5] : [1,4]) + IT([3,7] : [3,11]) + IT([2,4] : [2,6]) = IT \text{ C.F.}$
7.  $IT([1,5] : [1,4]) + IT([3,7] : [3,14]) + IT([2,7] : [2,6]) = IT \text{ C.F.}$
8.  $IT([1,5] : [1,6]) + IT([3,9] : [3,7]) + IT([2,2] : [2,6]) = IT \text{ C.F.}$
9.  $IT([1,5] : [1,6]) + IT([3,9] : [3,8]) + IT([2,5] : [2,6]) = IT \text{ C.F.}$
10.  $IT([1,5] : [1,6]) + IT([3,9] : [3,11]) + IT([2,4] : [2,6]) = IT \text{ C.F.}$
11.  $IT([1,5] : [1,6]) + IT([3,9] : [3,14]) + IT([2,7] : [2,6]) = IT \text{ C.F.}$

Où C.F. =  $([1,5] : [2,6])$

Il existe une infinité de solutions dans la répartition des tolérances, qui sont compatibles avec le système d'équations sus mentionné. Une façon simple peut être appliquée pour

répartir l'intervalle de tolérance sur toutes les cotes de la chaîne. Cette façon consiste à diviser l'intervalle sur le nombre des cotes (Équation 6.2).

$$ITC_{ij} = \frac{ITJ_i}{NC_i} \quad 6.2$$

- $ITC_{ij}$  : Intervalle de tolérance de la cote entre deux éléments (i et j)
- $ITJ_i$  : Intervalle de tolérance de la condition fonctionnelle entre deux éléments
- $NC_i$  : Nombre des cotes dans la chaîne des cotes.

L'équation 6.2 s'applique à des chaînes de cotes unidirectionnelles (1D). Mais par contre, nous pouvons élargir la fonctionnalité de cette équation pour l'appliquer à nos chaînes obtenues (à 3D) et ceci en déterminant les projections dans les trois plans standard pour obtenir trois équations du même type que 6.2 pour chaque chaîne obtenue par notre programme.

Celle-ci est une solution rapide mais qui a cependant l'inconvénient de nier les difficultés de réalisation des différentes pièces. Le problème se complique si l'on désire attribuer à chaque cote un intervalle de tolérance rationnelle, tenant compte des moyens de réalisation et de contrôle.

### 6.2.2 Analyse de l'influence de chaque chaîne

Une autre utilisation de l'arbre des éléments fonctionnels obtenu par notre programme est de déterminer les chaînes de cotes les plus influentes autour d'une condition fonctionnelle. Ceci peut se faire en appliquant les règles proposées par L. Laperrière et P. Lafond [41].

L'identification des chaînes les plus influentes peut être réalisée par l'application des deux règles suivantes :

- ♦ *Élimination des chaînes qui ne correspondent pas à l'exigence fonctionnelle recherchée;*
- ♦ *Conservation des chaînes qui possèdent les plus de paires d'éléments fonctionnels.*

L'application de ces règles peut se faire en appliquant des conditions dans notre algorithme au moment de l'affichage pour vérifier le nombre de paires dans chacune des chaînes (une simple boucle pour compter le nombre de fois qu'une paire se répète dans une chaîne).

Nous pouvons déterminer aussi les paires les plus influentes (paires qui se répètent le plus) dans un mécanisme en partant du même principe que celui utilisé pour l'obtention des chaînes les plus influentes.

### 6.2.3 Modélisation mathématique

Une fois les chaînes de cotes obtenues par l'algorithme d'identification des chaînes de cotes, il reste à affecter à chaque cote une zone de tolérance.

En utilisant la cinématique inverse, L. Laperrière et P. Lafond [41] proposent une approche pour modéliser les différentes dispersions des éléments fonctionnels affectant la ou les condition(s) fonctionnelle(s). Cette approche permet d'obtenir six équations à  $n$  inconnues (Équation 6.3) où  $n$  est le nombre de paires d'Efs. Pour plus de détails, il serait préférable de consulter la référence source.

La méthode Jacobienne se base sur le fait que l'on a des petits déplacements. Puisque nous travaillons dans un espace tridimensionnel, il est possible d'exprimer la position d'un point par rapport à un autre à l'aide de trois translations et trois rotations (ces points peuvent être assimilés à des joints à six degrés de liberté) pour obtenir les six équations à  $n$  inconnus.

La modélisation mathématique relative aux six petits déplacements des paires d'éléments fonctionnels autour de la condition fonctionnelle à l'intérieur d'une chaîne de tolérance est donnée par la formule (6.3)

$$\begin{bmatrix} \vec{\delta S} \\ \vec{\delta \alpha} \end{bmatrix} = \left[ (J_1 J_2 \dots J_6)_{FE_1} \dots (J_1 J_2 \dots J_6)_{FE_n} \right] * \begin{bmatrix} \vec{\delta q_{FE_1}} \\ \dots \\ \vec{\delta q_{FE_{n-1}}} \\ \vec{\delta q_{FE_n}} \end{bmatrix} \quad 6.3$$

Où

- $\vec{\delta S}$  : Trois vecteurs de petites rotations
- $\vec{\delta \alpha}$  : Trois vecteurs de petites translations du point d'intérêt
- $(J_1 J_2 J_3 J_4 J_5 J_6)_{FE_i}$  : Matrice Jacobienne 6x6 associée à une paire d'éléments fonctionnels tolérancé de la  $i^{ème}$  paire.
- $\vec{\delta q_{FE_i}}$  : Vecteurs de petits déplacements associés à l'élément fonctionnel tolérancé de la  $i^{ème}$  paire

L'obtention des matrices Jacobiennes est réalisable en utilisant les résultats obtenus par le programme (Algorithme d'identification des chaînes de cotes). Il faut tout simplement affecter les matrices Jacobiennes de l'équation 6.3 par les chaînes de cotes obtenues auparavant.



## 6.2.4 Analyse de robustesse du design

L'algorithme d'identification des chaînes de cote permet une extension possible pour l'analyse de la robustesse d'un design mécanique.

La robustesse générale d'un concept est augmentée en réduisant au minimum le nombre de paramètres de contrôle pour la caractéristique critique et en maximisant la capacité générale de supprimer la variation. Selon *Rikard Söderberg* et *Lars Lindkvist* la robustesse est définie par "La capacité maximale de supprimer les données de la variation géométrique". La variation est la variation réelle d'une surface particulière ou d'un dispositif géométrique d'un design. [48]

Les différentes étapes du processus de conception se divisent en deux sous étapes :

1. Pendant la conception de l'avant-projet de la géométrie, où des conditions fonctionnelles sont remplies par la sélection des caractéristiques de la géométrie, l'information minimale de fabrication est disponible et les efforts devraient porter sur l'amélioration de la robustesse de la conception.
2. Pendant la conception de détail de la géométrie, un concept de construction remplissant les conditions fonctionnelles du produit doit être ajusté sur les contraintes réelles de fabrication. Ceci signifie; compenser la sensibilité géométrique du concept par la sélection des tolérances et réduire au minimum le coût de fabrication du produit.

### 6.2.4.1 Axiomes de robustesse

Dans la théorie de conception axiomatique[48], l'équation 6.4 associe un groupe de paramètres de conception parallèles (DP) qui constitue des dispersions des paires d'éléments fonctionnels. Les éléments de la matrice représentent les dérivées partielles à des points spécifiques qui sont ici les paires d'éléments fonctionnels. La partie gauche de l'équation représente les dispersions de la condition fonctionnelle.

$$\begin{bmatrix} CF_x \\ CF_y \\ CF_z \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} DP_x \\ DP_y \\ DP_z \end{bmatrix} \quad 6.4$$

Une conception désaccouplée se caractérise par le fait que chaque sortie est contrôlée par une entrée seulement. Une conception de ce type est représentée par une matrice diagonale dont les éléments non-diagonaux sont égaux à zéro [48]. À la lumière de cette relation il ne peut exister qu'un lien entre l'entrée et la sortie.

Nous allons discuter dans la section suivante, comment cette philosophie peut être appliquée aux problèmes de tolérancement dans lesquels le concept géométrique peut être amélioré en diminuant le degré géométrique de couplage, c.-à-d. le nombre d'entrées à contrôler un certain résultat.

#### 6.2.4.2 Application au tolérancement

Nous pouvons augmenter la *robustesse* d'un design mécanique à partir de l'algorithme de l'identification des chaînes de cotes en minimisant le nombre des contrôles dans ce mécanisme. La minimisation de ces contrôles peut se faire en minimisant le nombre des conditions fonctionnelles.

Un design mécanique ne doit pas comporter deux conditions fonctionnelles dans une seule chaîne de cotes. L'exemple 3.1 contient deux conditions fonctionnelles : la deuxième est incluse dans la première, ce qui implique que le design n'est pas robuste.

L'algorithme développé donne les chaînes autour d'une condition fonctionnelle. On peut obtenir la liste de toutes les chaînes de cotes autour de toutes les conditions fonctionnelles dans le mécanisme en question.

Une fois cette liste connue, il suffit d'implanter une fonction, au niveau de l'obtention de l'arbre des éléments fonctionnels, dont le rôle est de vérifier si une chaîne est incluse dans une autre, et de vérifier si une condition fonctionnelle est incluse dans une autre.

### 6.3 LIMITES ET RECOMMANDATIONS

L'algorithme ici proposé détecte toutes les chaînes de cotes autour d'une condition fonctionnelle sélectionnée par l'utilisateur. Toutefois cet algorithme a ses limites. Ces dernières sont décrites ci-dessous, avec pour chacune une recommandation pour tenter de la contourner.

- ◆ Les sélections des contacts et des conditions fonctionnelles entre différentes pièces sont effectuées manuellement par l'utilisateur. Dans la version actuelle, ces sélections sont difficiles car elles exigent une interprétation de la nomenclature en association avec les éléments géométriques à sélectionner. Une sélection des contacts et des conditions fonctionnelles directement sur les entités géométriques serait de beaucoup préférable.

Par exemple, la sélection des contacts et de la condition fonctionnelle pourrait se faire à l'écran en utilisant des couleurs différentes pour identifier les différents éléments fonctionnels sélectionnés sur les différentes pièces. Ceci peut se faire en identifiant les éléments fonctionnels et ensuite en modifiant la couleur dans le fichier BREP initial pour l'afficher à nouveau à l'écran.

Autre façon de faire est d'automatiser la détermination des contacts entre différentes pièces en utilisant l'approche proposée par Bre [14].

L'automatisation de la détermination des contacts entre différentes pièces peut se faire en comparant chaque élément fonctionnel de chaque pièce avec les autres éléments des autres pièces. L'existence d'un contact au moins entre les éléments de deux pièces entraîne un contact entre ces deux pièces en question.

- ◆ Pour obtenir les chaînes de cotes autour de plusieurs conditions fonctionnelles, l'algorithme proposé oblige l'utilisateur à sélectionner une condition à la fois. Il est préférable de sélectionner toutes les conditions en même temps, pour évaluer la robustesse du mécanisme. Ceci peut être réalisé en utilisant le même principe de sélection que celui utilisée pour les contacts.
- ◆ Les résultats obtenus sont difficilement interprétés (le système de numérotation dans le programme). Nous pouvons améliorer les résultats obtenus qui sont afficher à l'écran en utilisant des noms significatifs pour les éléments fonctionnels et les solides à la place des chiffres.
- ◆ L'implantation de l'algorithme ne tient pas compte de la minimisation de la mémoire. Il faudrait vérifier s'il y a redondance dans le code et éliminer toute redondance.

## Références

---

- [1] A. Kaufman «Introduction à la combinatoire en vue des applications » *DUNOD Paris* 1968
- [2] A. A. and G. Requicha , « Toward a theory of geometric tolerancing », *the internat journal of robotics resarch*, 2(4). PP. 45-6-. Winter 1983.
- [3] A.A. and G. Requicha and S. C. Chan, «Representation of geometric features, tolerances, and attributes in solid modelers based on constructive geometry», *IEE Journal of Robotics Automation*,v 2,n3,pp156-166, 1986
- [4] Ann B.N.K. and Chuan K.Y., , « Conceptual Development of an Automated Assemblies Tolerance Analysis System », *International Jornal of Computer Applications in Technology*, v9, n1, PP9-17. 1996
- [5] André Clément, Alain Rivière et Philippe Serré, «Les SATT : un modèle déclaratif intégré pour la paramétrisation, le tolérancement et l'assemblage », *ISMCM*, 1996
- [6] A. Desrochers and A. Clément , , «A dimensioning and tolerancing assistance model for CAD/CAM systems », *Int. J. of Advanced Manufacturing Technology*, v9, PP352-361, 1994
- [7] A.A.G. Requicha « Toward a theory of geometric tolerancing », *the International Journal of Robotics Research*, v2, n4, pp45-60, 1983
- [8] A.A.G. Requicha « Mathematical meaning and computational representation of tolerance specification Proc ». *Int. Forum on Dimensional Tolerancing and Metrology*, Dearborn, Michigan, CRTD - v27, pp. 61-68, June 1993.
- [9] A. Derochers and R. Maranzana, « Constrained dimensioning and tolerancing assistance for mechanisms » *4<sup>th</sup> CIRP Seminar on Computer Aided Tolerancing*, The university of tokyo, Japan, april 5-6, 1995
- [10] Brayn Kok Ann Ngoi and Yew Chaun Kuan, , « Tolerance charting : the state-of-art review » *International Journal of Computer Application in Technologie*, v8, v3/4, pp229-242. 1995
- [11] Brintton G.A. Whbreew K. and Tor S.B., , « An Industrial Implementation of Computer-Aided Tolerance Charting », *International Jornal of Advanced Manufacturing Technology*, v12, n2,pp122-131. 1996
- [12] Brintton G.A. Whbreew K. and Tor S.B. « Operational dimensioning in process planning setup planning », *Int. J. of PROD*, v34, n7,pp1841-185822-131, 1996

- [13] B. Charles, A. Clément et A. Rivière. «Les SATT un modèle technologique pour la CFAO», *Technologie*, n80, pp71-77, 1992
- [14] Bre Moussa Diamande, «Détermination d'une séquence optimale par le regroupement des opérations », *MEMOIRE MAITRISE*, UQTR, 1995
- [15] Bre Moussa Diamande and Luc Laperrière «Automatic liaison model generation from 3-D solid models», *Transactions of the CSM*, v20, n4, pp333. 1996
- [16] C. Berge, « The Theory of Graphs and its Applications », *Wiley, New York*, 1973.
- [17] C. Berge, « The Theory of Graphes et Hypergraphes », 2<sup>e</sup> édition, *Dunod, Paris*, 1973.
- [18] C. Berge, «Graphes », *Gauthier-Villars, Dunod*, 3<sup>ème</sup> édition
- [19] Desrochers and A. Clément « Manuel Technologie » 1994
- [20] Frank HARARY & Robert Z. Norman «Introduction à la théorie des graphes orientés *modèles structuraux* » DUNOD Paris 1968
- [21] Ferruh Ozturk and kadir Cavdar, « A computer based approach to dimensioning and tolerancing », *Engineering Systems Design and Analysis* , v5, n15, p0143149-765, 1994
- [22] F. Boyer, «Modélisation volumique basée sur une représentation duale de la géométrie », *Général electric CAE international/SDRC*, n75, pp32-47, 1995
- [23] «Guide du dessinateur industriel » A. Chevalier , Édition 1988
- [24] *Giesecke micell spencer hill dygdon*, «dessin technique».
- [25] H. Riberrol, «Aide-mémoire, dessin industriel», *Delarative*, 1973
- [26] Ibrahim Coulibaly «Contributions à l'analyse numérique des méthodes quasi-Monte Carlo » *thèse de doctorat*, Université Joseph Fourier
- [27] J.F.Rigal, V.Wolff, B.Roumesy, S. Raynaud, , « Un modele géométrique pour le suivi de la cotation en cours de fabrication en CFAO », *First International Conferance* , n15, pp757-765, 1996
- [28] Johnson , R. H., « Dimensioning and toleancing final report », *Computer aided manufacturring international (CAM-I) USA*, 1985
- [29] K. M. Yu, S.T.Tan and M.F.Yuen , « A review of automatic dimensioning and tolerncing shemes », *Engineering Computers* , v10, pp63-80, 1994
- [30] Luc Laperrière. «Notes de cours : Tolérancement et Métrologie, ING-1045 », *UQTR*, 1997

- [31] Luc Laperrière, «Identifying and quantifying fonctionnal elements dispersions during functional analysis», *5EM CIRP International Seminear on Cumputer Aideid Tolerancing* , Toronto. 1997
- [32] L. Mathieu, R. Weill «A model for machine tool setting as a fonction of positionning errors » *Seminar on dimensioning and tolerancing*, CIRP, Pennstate, 5/1991
- [33] L. Mathieu «Concepts pour la spécification géométrique des pièces », *Journées cotation*, IUT Annecy, 1993.
- [34] L. Mathieu, Alex BALLU, « Un modèle pour la géométrie fonctionnelle des produits » LMP - Université Bordeaux I
- [35] Michel Temmerman., , «Cotation tridimensionnelle des systèmes mécaniques», *Technologie*, n75, pp32-47, 1995
- [36] Mohamed Noui Milki, Daniel Mennier , «Dimensioning and functional tolerancing aided by computer in CAD/CAM systems –Application to Autocad system», *IEEE*,1995
- [37] Michel Godon and Michel Minoux «Graphes et Algorithmes » *EDITIONS EYROLLES* 1979
- [38] Marti Mantyla , « An introduction to solid modeling », *COMPUTER SCIENCE PRESS*, 1988
- [39] Owen Ransen, Autocad programming in C/C++, *WILLEY*,1997
- [40] Philippe Vanackere, «Les modélisations géométrique utilisées dans les logiciels de CAO», *Technologie*, n50,pp51-54, 1992
- [41] Philippe Lafond, «Proposition d'une méthodologie pour l'identification systématique des dispersions affectant une condition fonctionnelle», *cour séminaire de recherche à l'UQTR, H-1996*.
- [42] Roland Maranzana, « Notes de cours : Système manufacturier,SYS-822 »
- [43] R. Ropion, « La cotation fonctionnelle des dessins techniques » , *Dunond*, 19768.
- [44] Serge Samper, Max Giordano, , « Modèles pour la prise en comptes des déformations dans la démarche de tolérancements», *First International Conferance* , n15, pp757-765, 1996
- [45] Tasseré Kaboré «Simulation de l'impact des dispersions d'éléments fonctionnels sur une condition fonctionnelle d'un assemblage mécanique. », Cours séminaire de recherche, U. Q. T. R.,1999

- [46] U Roy, C Rliu and T C Woo. , « Review of dimensioning and tolerancing : representation and processing », *CAD*, v23, n7,pp466-483,1990
- [47] V. Srinivasan « The role of sweeps in tolerancing » *semantics Manufacturing Review* , Vol. 6, No. 4, pp. 275-281, 1993.
- [48] R. Söderberg, Lars Lindkvist Tow-Step «Procedure for Robust Desgn Using CAT Technology », *Proceedings of the 6<sup>th</sup> CIRP, Enshede*, Global Consistincy of Tolerances, No. 4, pp. 231-240, 1999.

## Bibliographie

---

- B. Charles, A. Clément et A. Rivière, 1992, «Les SATT, un modèle technologique pour la CFAO », *Technologie*, n50, pp71-77.
- Genba Zhang and Michel Prochet , 1993, « Some new developments in tolerance design in CAD» *Advances in Design Automation*, V2, 65-2 DE-VOL
- René Pauliac , 1992, « Gérer la géométrie fabriquée», *Technologie*, v75
- Gérard T. , 1995,«L'optimisation en cotation fonctionnelle» *Technologie*, v75, PP9-23
- Uptal Roy and C. R. Liu, 1988, « Feature-based representational scheme of solid modeler for providing dimensioning and tolerancing information», *Robotics & Computer-integrated Manufacturing*, v4,n3/4, pp 335-345
- G. Abou and R. Cheng , 1994, « TVCAPP, tolerance verification in computer-aded process planing», *INT. J. PROD. RES*, v31,n2,pp393-411
- Jami J. Shah and Bing-Chum Zhang ,92,« Attributed graph model for geometric tolerancing», *Advances in Desgin automation*, v2, n44-2.
- Ferrth ozturk and Kadir Cavdar ,94, « A computer based approach to dimensioning and tolerancing», *Enginneering Systems Design and Analysis* , v2, 64-5
- Partt, M. J. 1984« Solid modling and the interface beteen design and manufacture», *IEEE Computer Graphics and applications* , v4,n7,52-59
- Dougals C. Greenwood « Mechanical detail for product design » *McGRAW-HILL book Company*, New York Toronto London
- Lowell W. Foster , 1986, « GEO-METRICS the metric application of geometric tolerancing technique as based on ANSI Y14.5M-1982 » *pratyces, addison-wesley*
- Mantyla , M. A. « An introduction to solid modeling», *Computer graphics and image processin* , V19,N1,PP28-42
- W. Straber H. P. Seidel (Eds.) «Theory and pratice of geometric modling » *Springer-Verlag Berlin Heidelber New York London*.
- R. A. Earnshaw, « Theorical foundations of computer graphics and CAO» *Springer-Verlag, Berlin Heidelberg New York London Paris Tokyo*
- R. Jaffeux ,1966 « Le dessin fonctionnel de définition»,*Dunod*



- Bernard Lefort et Georges Landreau , 1993, « Dessin industriel » Ministère de l'éducation, service des cours par correspondance.V1 et V2
- R. Macheeret « Cours de dessin industriel », *Dunod*, V1,V2
- Proceedings manufacturing international 90
- Slaby, S. M. « THREE-DIMENSIONAL DESCRIPTIVE GEOMETRY » *New York : Harcourt*
- Thomaths E.French, Carl L. Svensen, Jay D. Hesel and Bayron Urbanik « *Mecanical drawing* » Tenth edition, *McGraw-HILL book company*.
- R. Durot, R. Lavaud et J. Visart,1971 «La cotation fonctionnelle principes et applications», *Classiques hachette*
- J. Bullerwell et Babarbey,1971 «Cours de cotation fonctionnelle , enseignement programmé », Fascicules 1,2,3. *Édition André Casteilla*
- Warren J. Luzadder«Fundamentals of engineering drawing for design product development and numerical control » Seventh edition
- Wim j. Teunissen and Jan va den Bos « 3D interactive computer graphics , the hieracical modeling system HIRASP »*Ellis Horwood Limited*,1990
- George Omma 1995 « Autocad 13 » *SYBEX*
- Serge Caminé 1995, « L'univers d'autocad 13 », *Édition Vermet Inc.*
- David Pitzer, Francis Soen, Howard Fulmer ,1996, « Autocad 13 pour windows 95, NT, 3.1 », *Macmillan*
- [ISO 1101] "Dessins techniques - Tolérancement géométrique - Tolérancement de forme, orientation, position et battement - Généralités, définitions, symboles, indications sur les dessins", ISO, 1983
- [ISO 1660] "Dessins techniques - Cotation et tolérancement des profils", ISO, 1987
- [ISO 5458] "Dessins techniques - Tolérancement géométrique - Tolérancement de localisation", ISO. 1987
- [ISO 5459] "Dessins techniques - Tolérancement géométrique - Références spécifiées et systèmes de références spécifiées pour tolérances géométriques", ISO. 1981

- [ISO 8015] "Dessins techniques - Principe de tolérancement de base", ISO, 1985
- [ISO 10303] ISO/WD 10303-47 N247, "Industrial automation systems and integration - Product data representation and exchange - Part 47: Integrated generic resources: Shape variation tolerances"